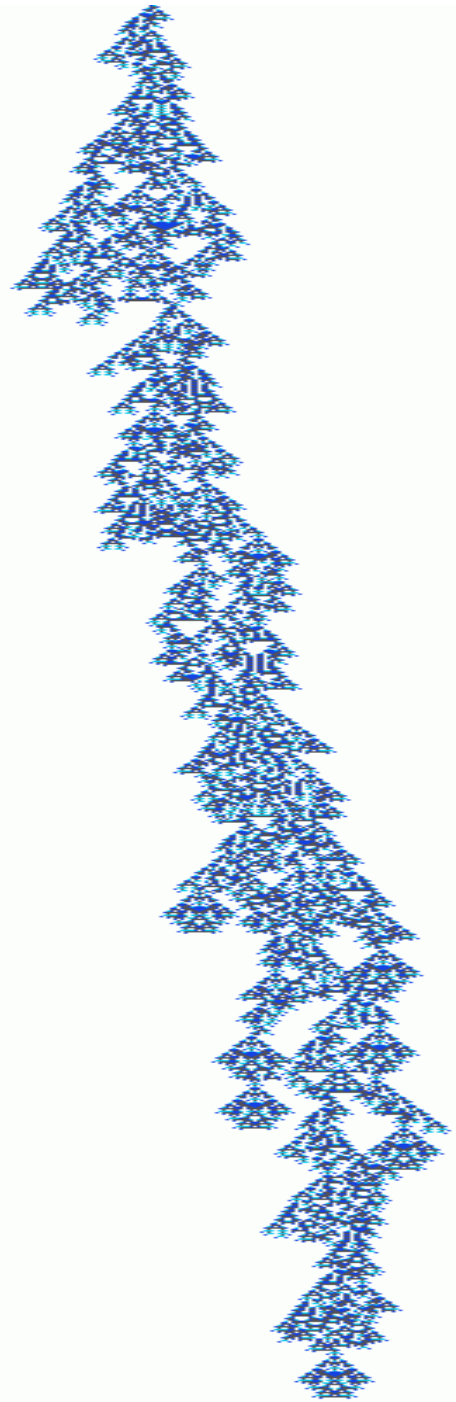
Cellular Automata

Franco Zambonelli
February 2010



Outline

- Part 1: What are CA?
  - Definitions
  - Taxonomy
  - Relevant Examples
- Part 2: CA as Complex Systems
  - Classes of Behavior
  - "The edges of chaos"
- Part 3: Non traditional CA models
  - Asynchronous CA
  - Perturbed CA
- Part 4: Applications of CA
  - Simulation
  - Implications for modern distributed systems

Part 1

- What are CA?

Why CA?

- CA are the simplest complex dynamical systems
  - Simple: easy to define and understand
  - Complex: exhibits very complex behaviors
- Very useful to
  - Understand complex systems
  - Grasp the power of interactions
  - Visualize complex behaviors
  - Simulate complex spatial systems

What are CA: Static Characteristics

- Discrete Dynamical System
- $$CA = (S, d, N, f)$$

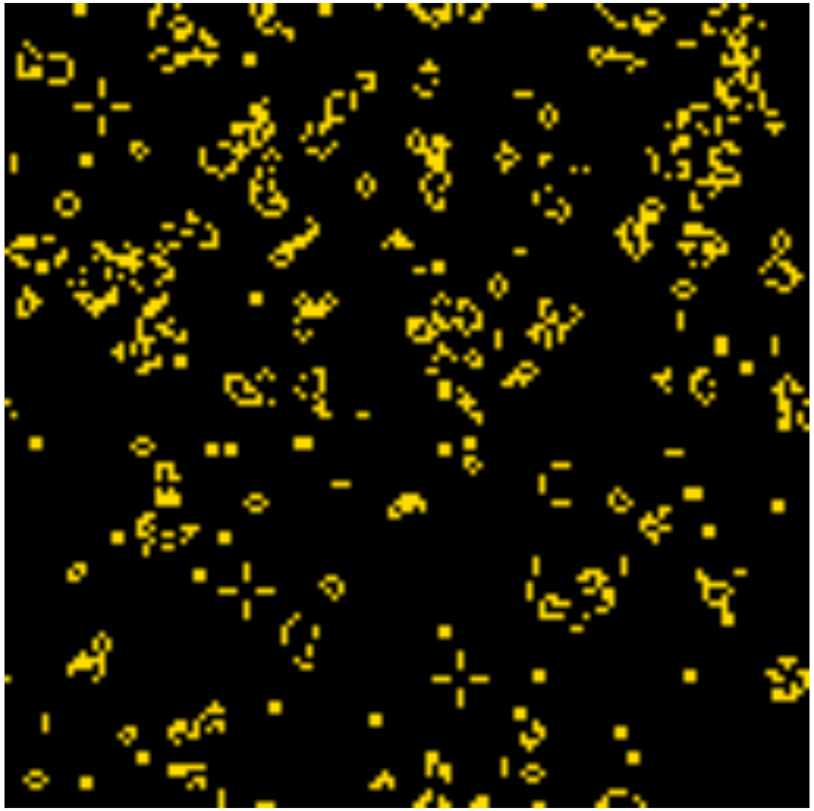
- Finite number of cells
  - Interacting in a regular lattice (grid-like, hexagonal, etc.)
- d = dimensional organization of the lattice
  - 1-D, 2-D, 3-D, etc…
- S = Local state of cells
  - Each cell has a local state
  - The state of all cells determines the global state
- f = State Transitions
  - Starting from an initial state S_i
  - Cells change their state based on their current state and of the states of neighbor cells
- N = Neighborhood
  - Which neighbor cells are taken into account in local state transitions

What are CA: Dynamic Characteristics

- Dynamic Evolution
  - Starting from an initial local state
  - Cells change their state based on their current state and of the states of neighbor cells
- Local vs. Global State
  - $S_{global} = (S_0, S_1, S_2, \ldots S_N)$
  - The Global state evolves as the local states evolves…
- Dynamics of state transitions
  - Synchronous
    - All cells change their state at the same time
    - For each cell i $S_i(t+1) = f(S_i(t), S_k(t) \ldots S_h(t))$
    - That is, the global state evolves in a sequence of discrete time steps
  - Asynchronous
    - Cells change their state independently
    - Either by scanning all cells
    - Or by having each cell autonomously trigger its own state transitions
  - Stochastic
    - For both Synchronous and Asynchronous CA, one can consider probabilistic rules for state transitions

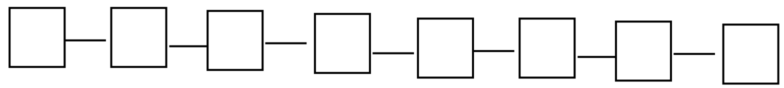
Visualizing CA Evolution

- The success of CA is due to the fact that
  - They are simple complex systems
  - They and their dynamic behavior can be visualized in a very effective manner
- Visualization
  - Associate to each local state a "color" (e.g., for binary states, "black" and "white")
  - Draw the CA grid and see how color changes as the CA evolves…

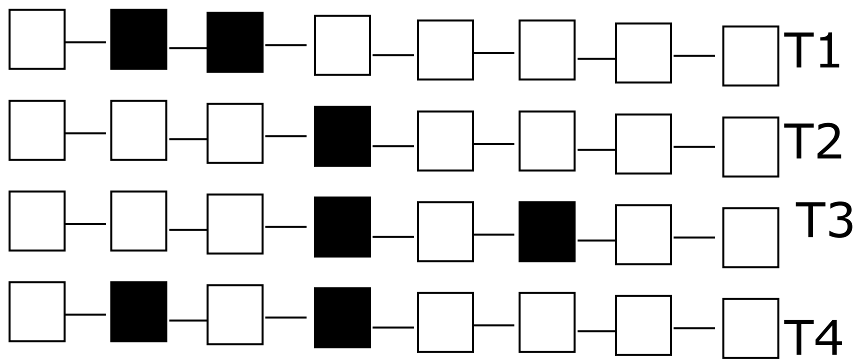


Structure of the Lattice: 1-D CA

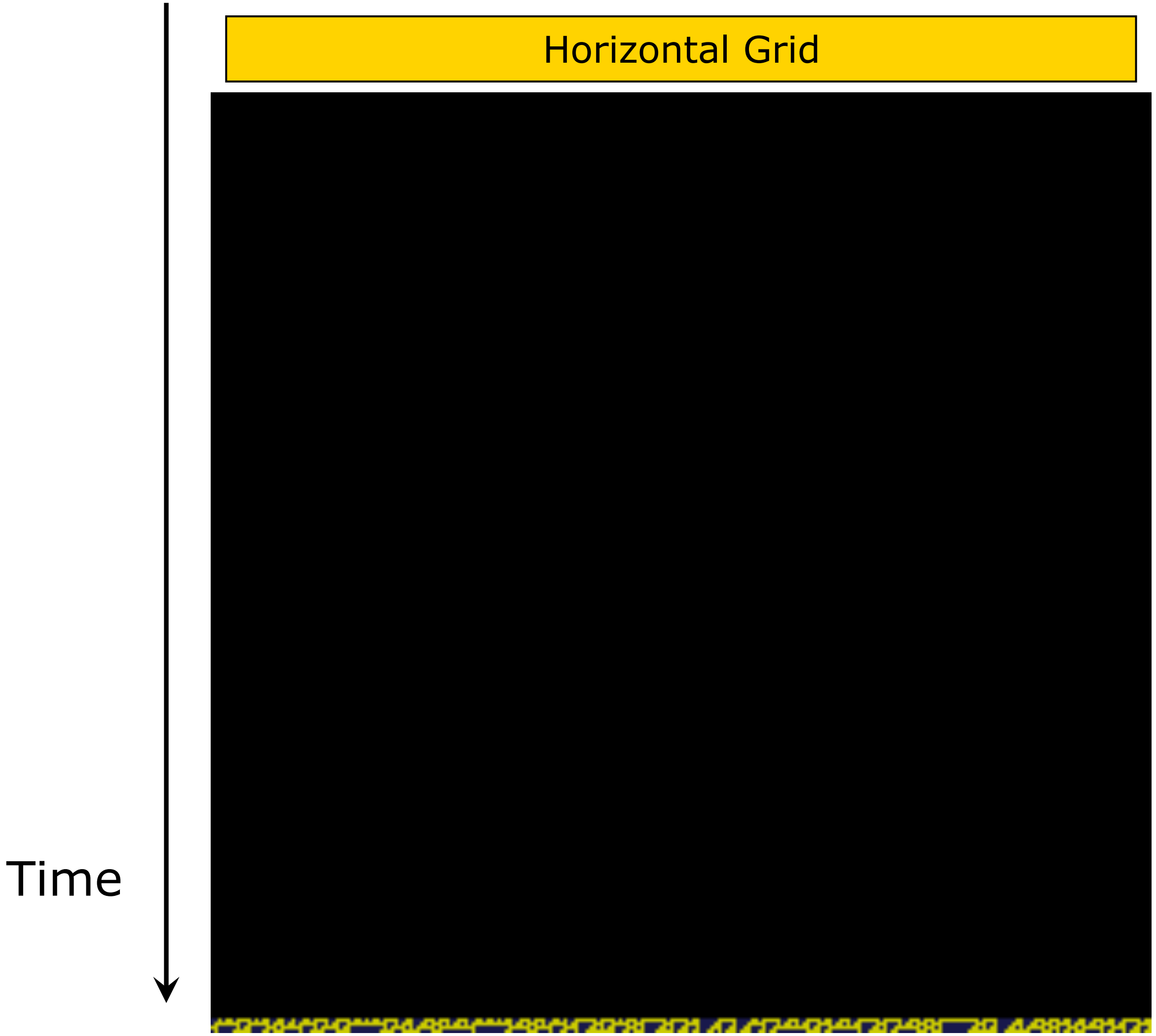
- Cells are placed on a straight line
  - connected each other as in a chain



- The evolution of a CA visually represent the evolution in time
  - How the configuration of the global state evolves step after step
  - In a single graph with time descending



Example of 1-D CA Evolution

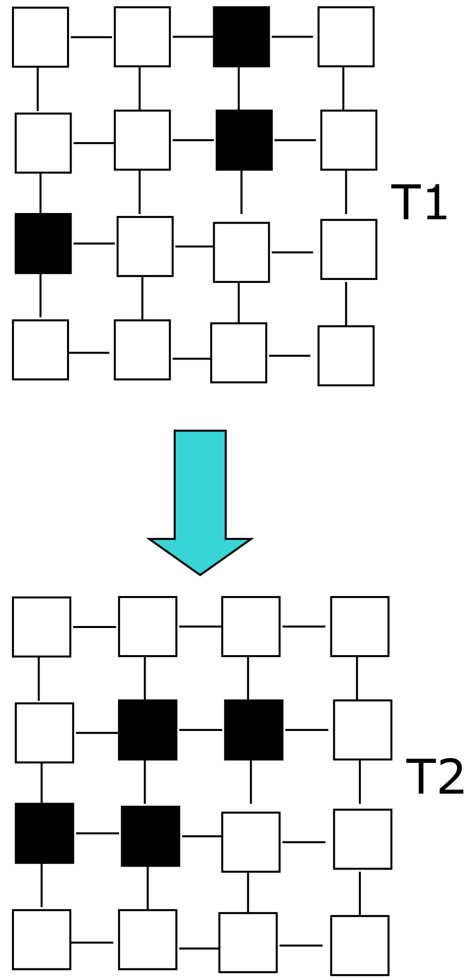


Each line of this grid represents a different time

This is clearly applicable only to synchronous CA, otherwise there would not be "global" time frames

Structure of the Lattice: 2-D CA

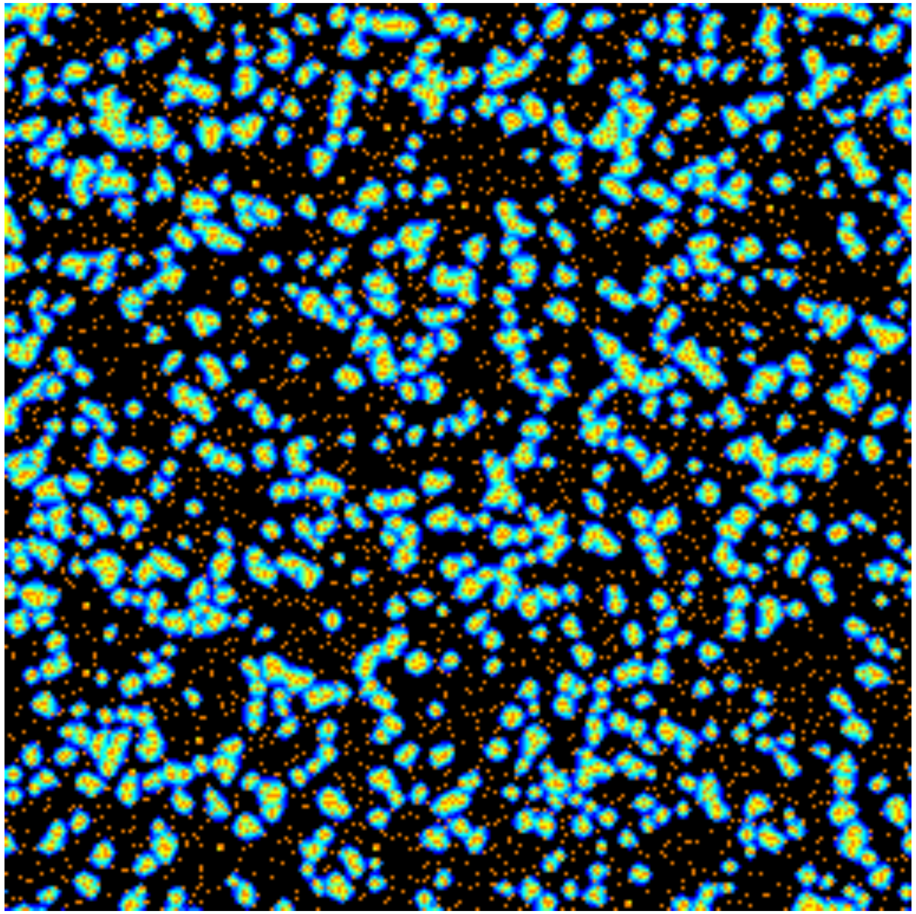
- Cells are placed on a regular grid typically mesh
  - but hexagonal “bee nest” structures are used too
- The evolution of a CA visually represent the evolution in space
  - How the spatial configuration of the global state changes



Example of 2-D CA Evolution

X Axis of the Grid

Y Axis of the Grid



The Time dimension is not explicitly visible

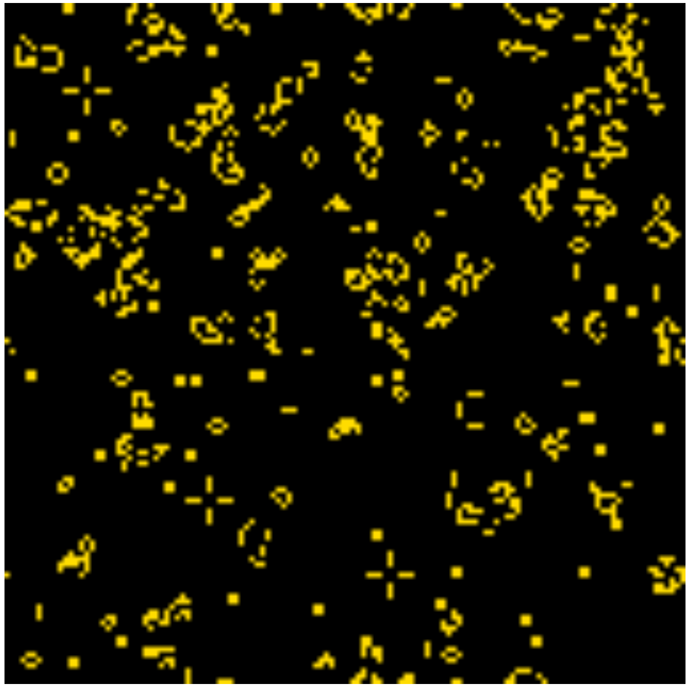
Local States of Cells

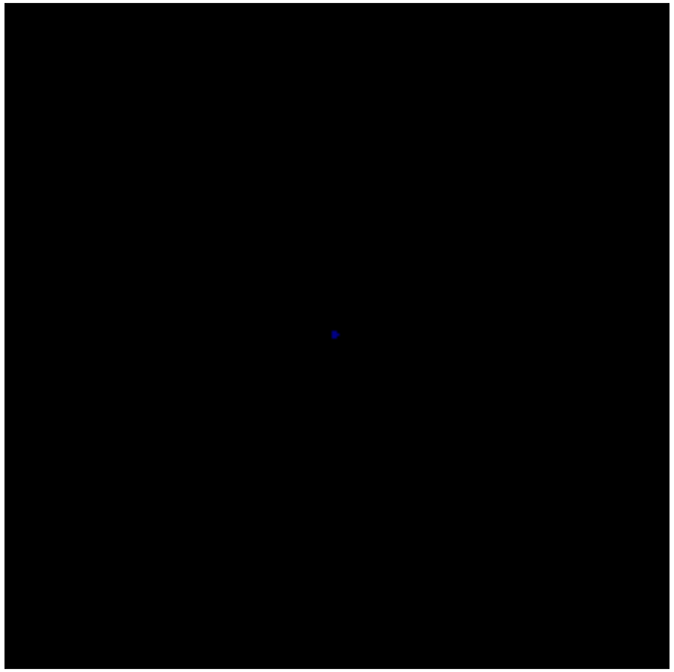
- Bynary state – simple but very used
  - Each cells has only two states
  - 0 – 1
  - “dead” or “alive”
  - “black” or “white”
- Finite state sets
  - Three, four, five, etc. states.
- Continuous states
  - The state varies over a continuous domain (i.e., the real axis)
  - Often represented as color shades
- Structured states
  - The state can be a “tuple” of values

- The choice depends of what one has to analyze/ simulate

Binary vs. Continuous State

- Examples for two different 2-D CAs





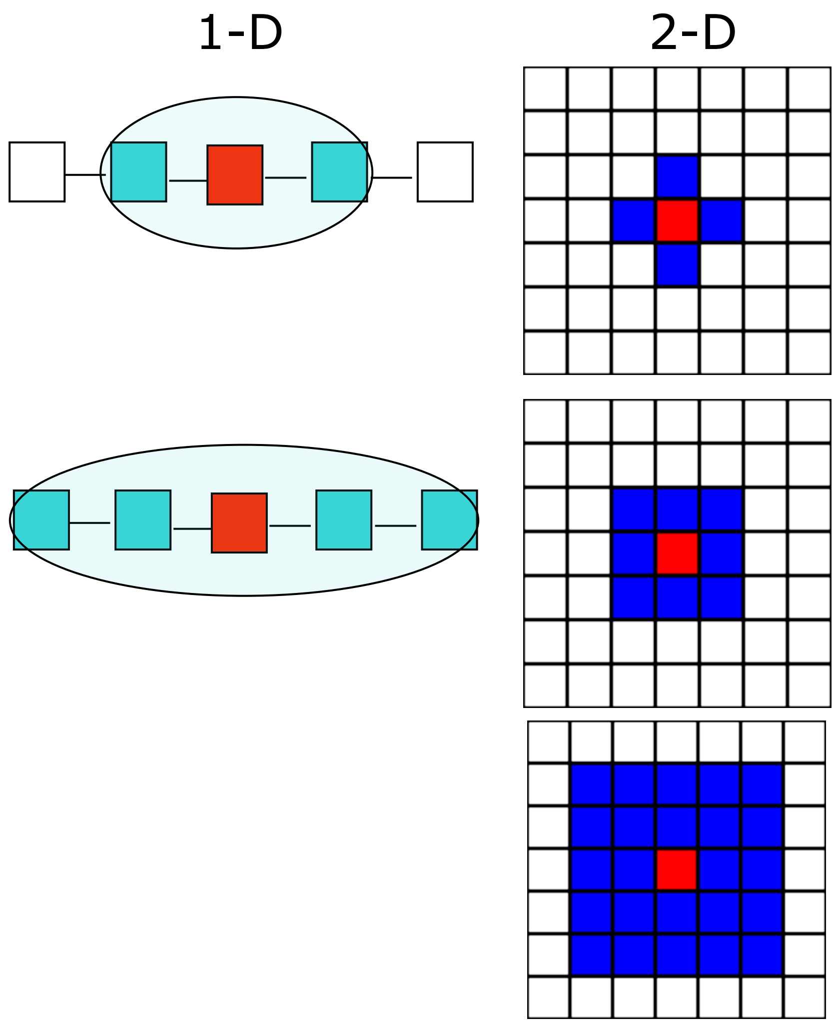
State Transitions

- In general, a cell compute its new state based on
  - Its own current state
  - The current state of a limited number of cells in the neighborhood
- Examples
- For a binary CA
  - *f = {a cell in state S=0 move to state S=1 iff it has 2 neighbors in state S=1; a cell in state S=1 stays in that state iff it has 1 or 2 neighbors in state S=1}*
- For a nearly continuous state CA (*S=0-25*5)
  - *f = {NS=A(Sneighbors) }*
  - *f= {each cell evaluates its next state NS by comparing its current state CS and the average value A of neighboring cells: if |CS-A|<64 then NS = A, if |CS-A|>64 and A≥128 NS=255, if |CS-A|>64 and A<128 NS=0}*

- The state transition function clearly has a dramatic impact on the global evolution of the system

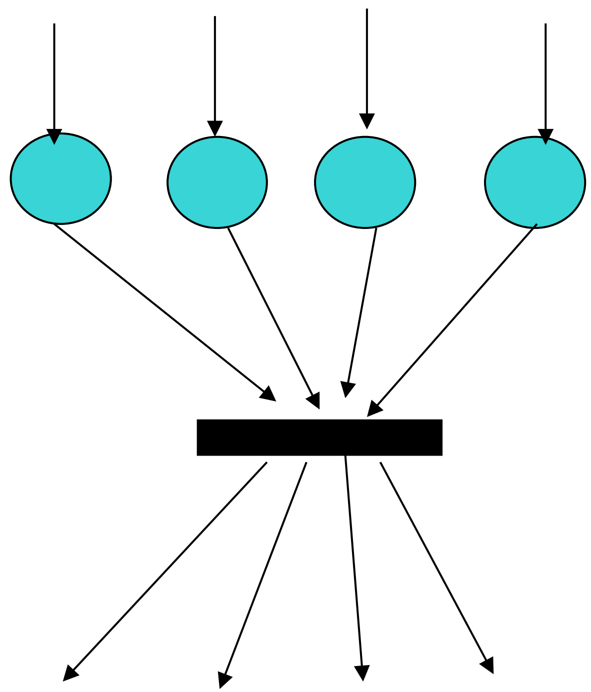
The Neighborhood structure

- How many neighbors are taken into account in state transition?
- Direct neighbors (dist = 1)
  - Only direct neighbor are considered
- Extended neighborhood
  - Neighbors at specific distances are considered
- Weighted neighborhood
  - The “impact” of a neighbor in state transitions decreased with distance (as in physical laws..)
- Non isotropic neighbor structures could also be considered
- All depends on what one has to study/simulate..



Synchronous Dynamics

- Cells change their state as in a synchronization barrier
  - All cells evaluate the transition rules with the local states at time T
  - All of them decide what their local state at time T +1 will have to be
  - All together pass to the next state together
  - Over and over for T+2, etc.
- SEE THE APPLET ON THE WEB!
- These are the mostly studied one, but do not always reflect the behavior of real-world systems



Asynchronous Dynamics

- In general
  - Each cell evaluate its current state and the current states of neighbor
  - And it performs its own state transition immediately
- However
  - There could be a global controller that decide which cells (and in which order) should do the evaluation (cfr. Object-based systems)
  - Or there could be no global controller, and each cell can autonomously trigger its own internal activities (Cfr. Agent-based systems)
- The behavior of Synch and Asynch CA can be very very different
- Asynch CA are more useful to study decentralized processes and to simulate decentralized systems

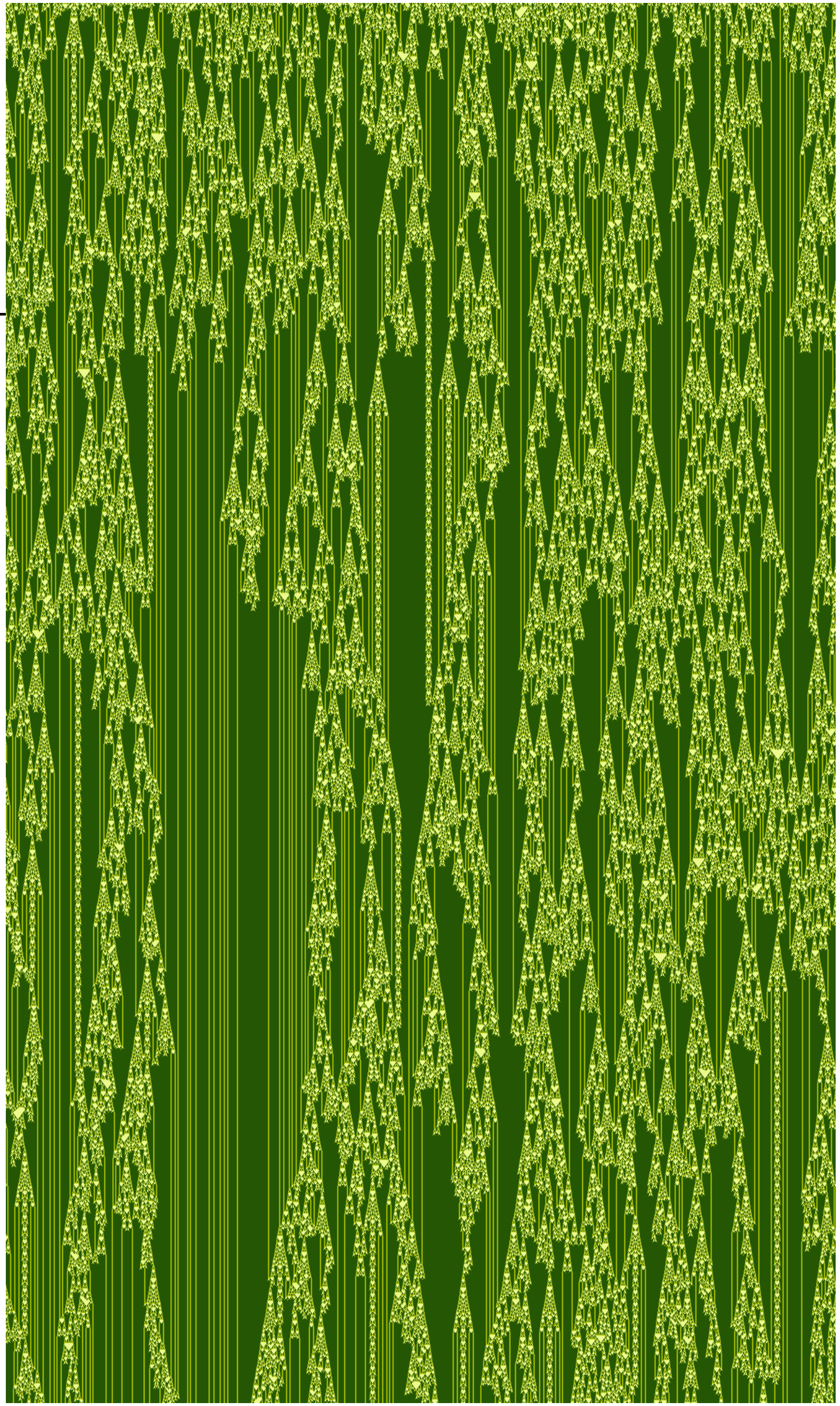
CA and Complex Systems (1)

- CA have all the ingredients that can lead to complexity
- It is a system of interconnected components
  - Even if very simply components
  - With very regular interaction topologies
- Local interactions
  - That influences the global behavior
  - Transitive effects → from local to global
- Feedbacks
  - The state of a cell influences its neighbors
  - The state of a cell is influenced by its neighbors
- Non linearity
  - NOT $S_i(t+1) = aS_h(t)+bS_k(t)$
  - BUT $S_i(t+1) = f_{nonlinear}(S_h(t), S_k(t),....S_j(t))$

- Openness and environmental dynamic?
  - Not necessary, but can be introduced to further increase the mess…

CA and Complex Systems (2)

- And CA indeed may exhibit very complex behavior
  - Self-organization (the edge of chaos)
  - Chaotic behavior
- Demonstrating “The Power of Interaction”
  - Very simple components and structure
  - Complexity emerges from interaction dynamics!



The CA “Hall of Fame”

- The Conway’s “Game of Life”
- The family of 256 binary 1-D CA rules
- Totalistic CA

The “Game of Life”

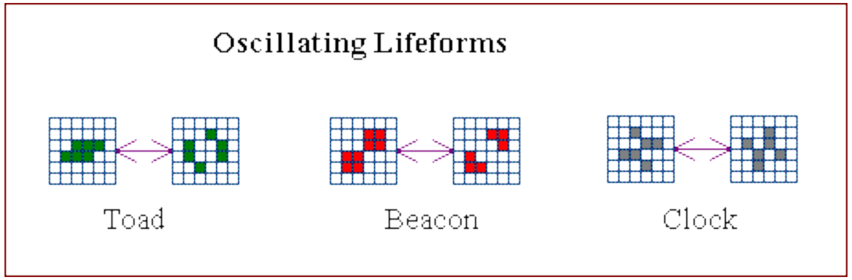
- Discovered by John Convey, mid ’70s
- 2-D, synchronous, binary CA
  - Cells are “dead” or “alive”
  - Seems like an ecosystems..
- Neighborhood
  - The Moore neighborhood
- Transition rule
  - *Birth*: a dead cell get alive if it has 3 alive neighbors
  - *Surviving*: an alive cell survive it is has 2 or 3 alive neighbors
  - *Death by overcrowding*: an alive cell dies if it has more than 3 neighbors alive
  - *Death by loneliness*: an alive cell dies if there are less than 2 neighbors alive

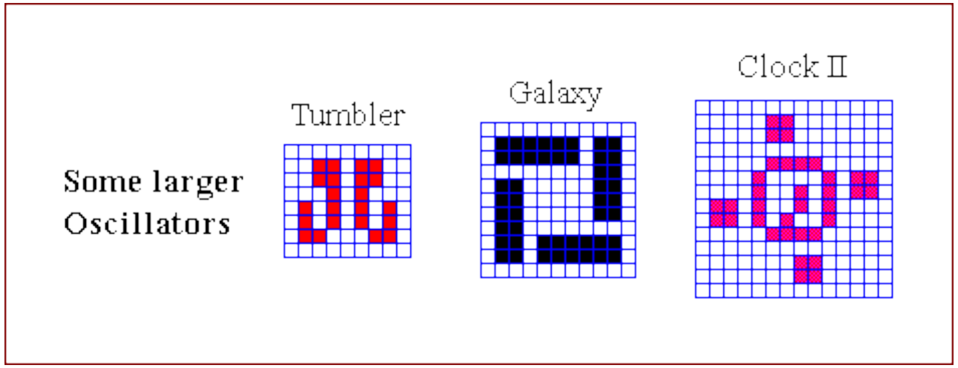
Game of Life: Why it is Interesting?

- The Game of Life shows very complex behaviors
  - In general, the evolution shows very complex patterns BUT
- There are stable structures
  - Stable ensembles of close cells that reach a equilibrium
- There are dynamic structures (“gliders”)
  - Ensembles of cells that survive by continously changing their shape
  - Ensembles of cells that roam in the world
- And there are complex interactions among these structures
  - Some structures annihilates others
  - Some structures generates others

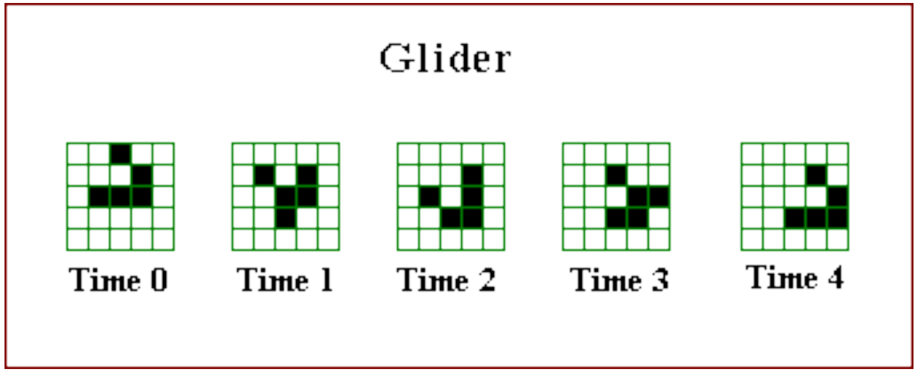
Game of Life: Life-forms Examples

- Oscillators



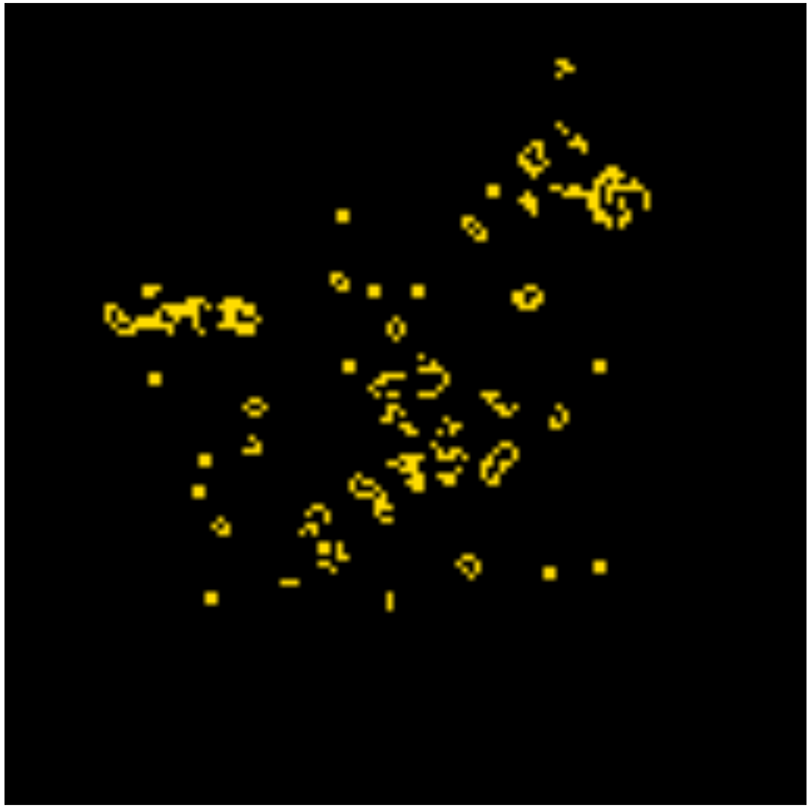


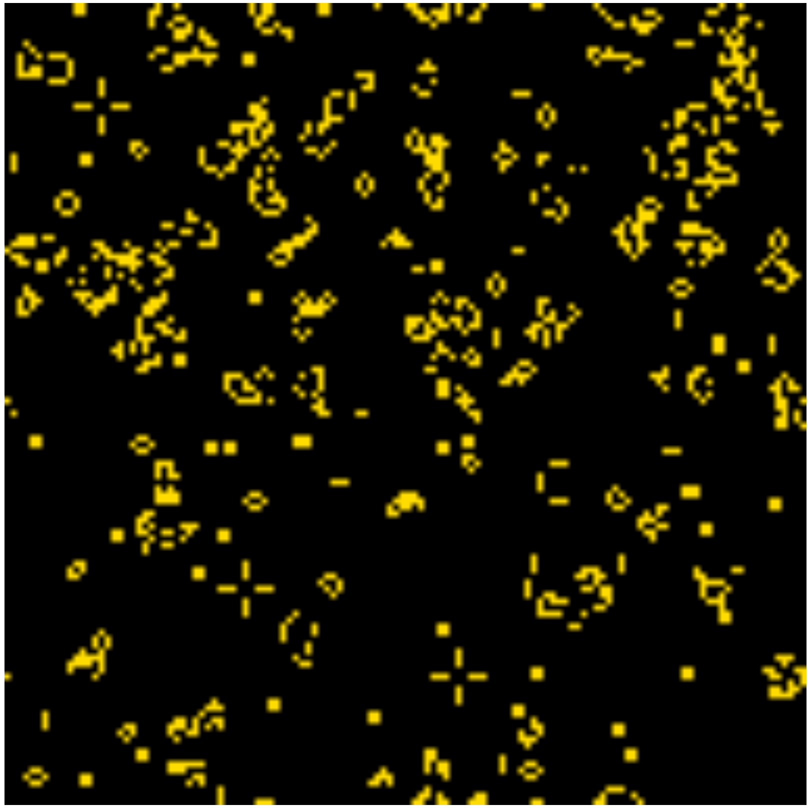
- Gliders



Game of Life: Some Snapshot

- See the world as it evolves…



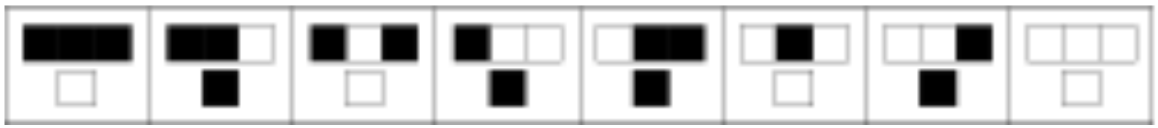


The 256 Elementary Binary Rules

- 1-D, binary, direct neighborhood
  - There are 256 possible rules
  - We can number rules in term of the binary number corresponding to the “next states list”

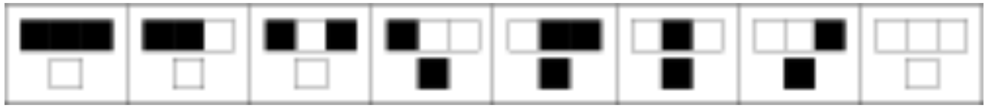
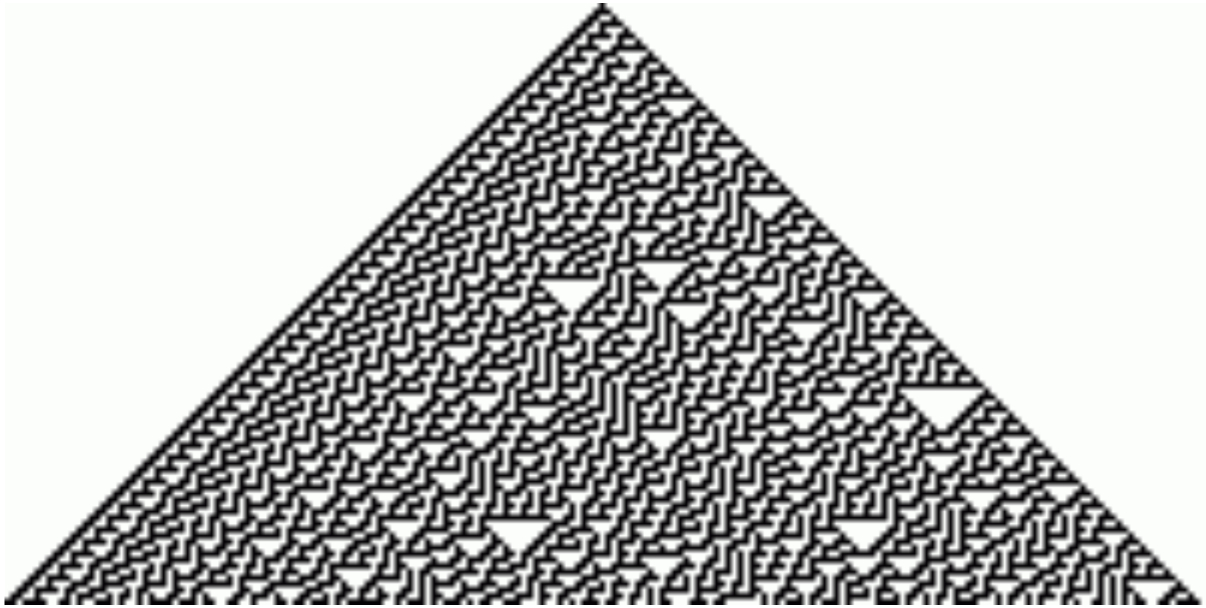
000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

- The above is rule “90”=01011010

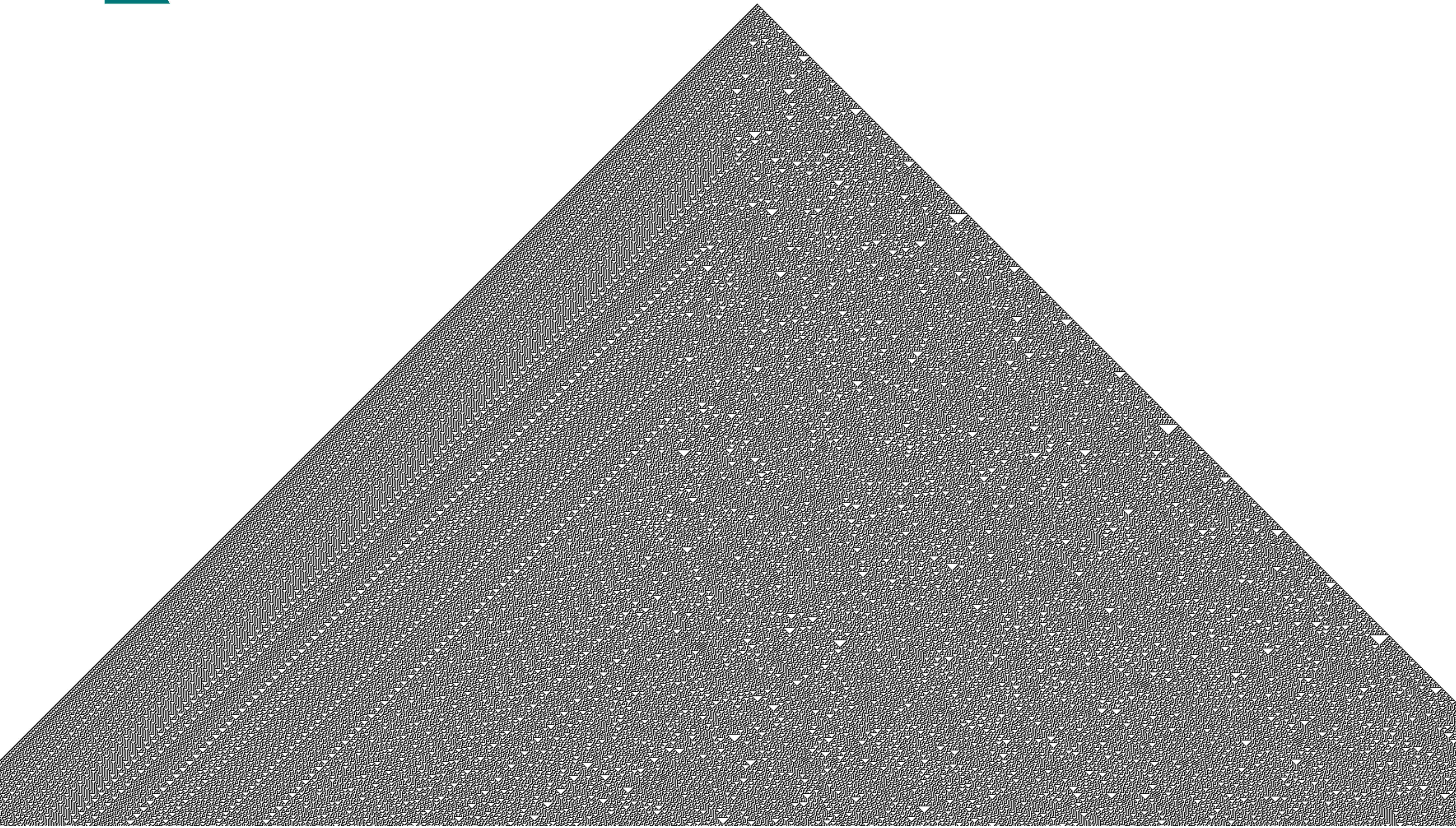


- We can then see how different rules evolve starting from
  - A random initial state
  - A single “seed”

Example: Rule 30

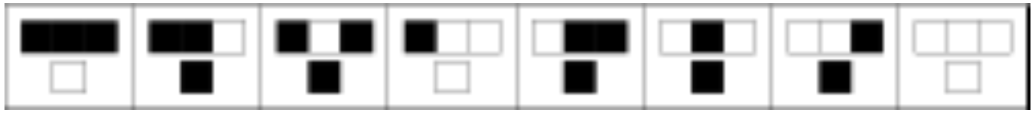
- Rules 
- Steps 
- Evolution 

More an Rule 30 Evolution…



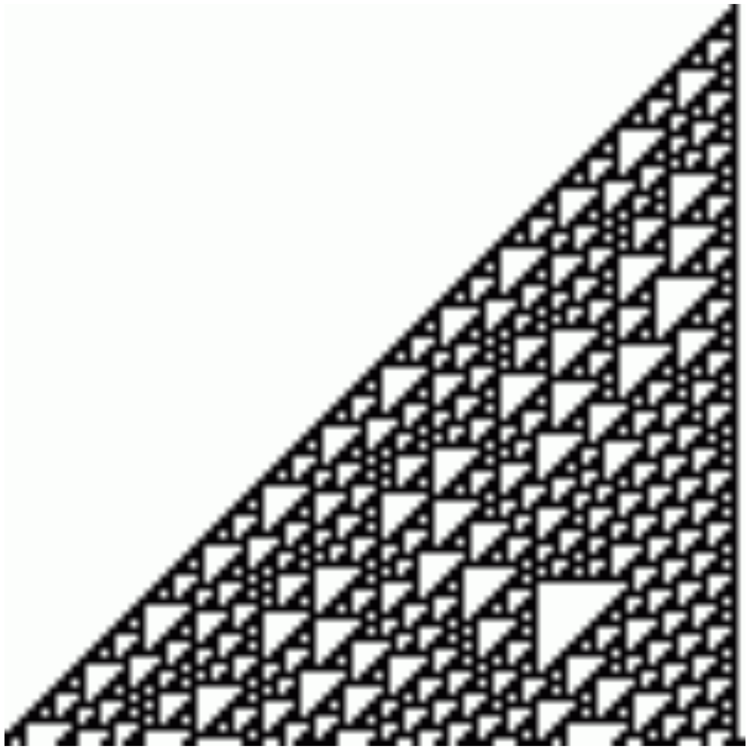
Example: Rule 110

- Rules

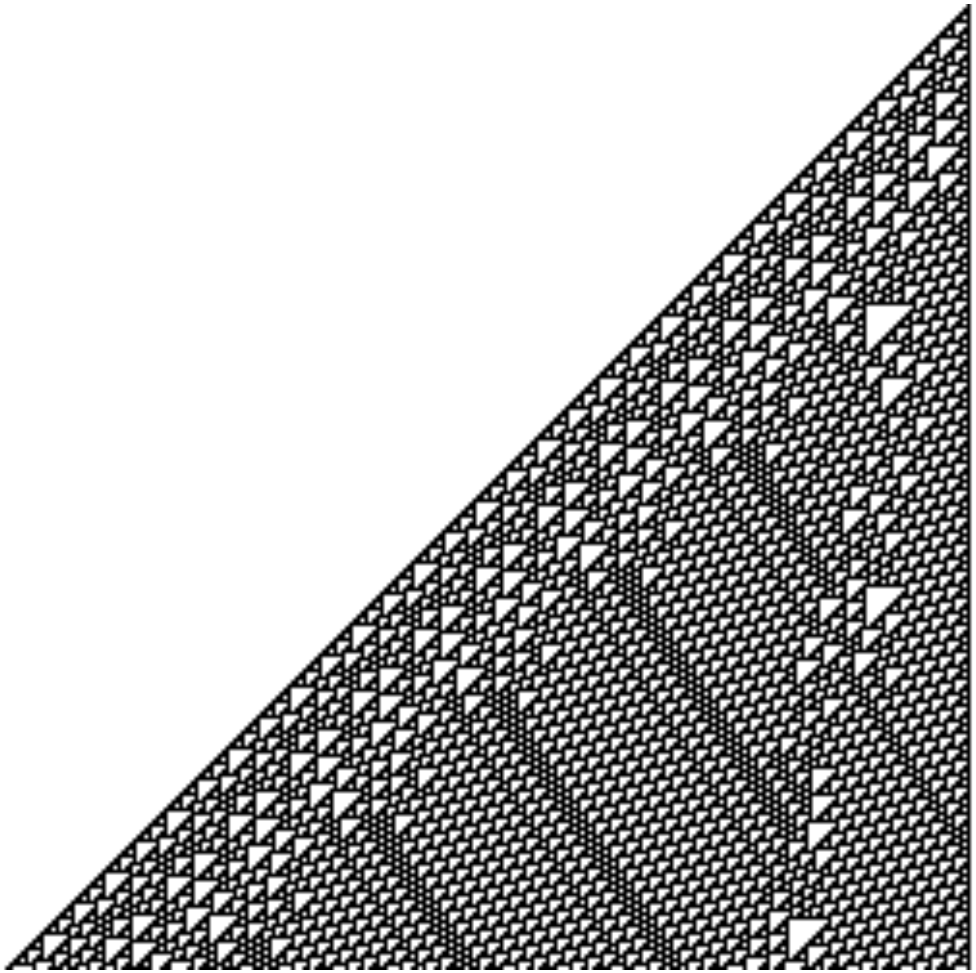


- Steps

- Evolution

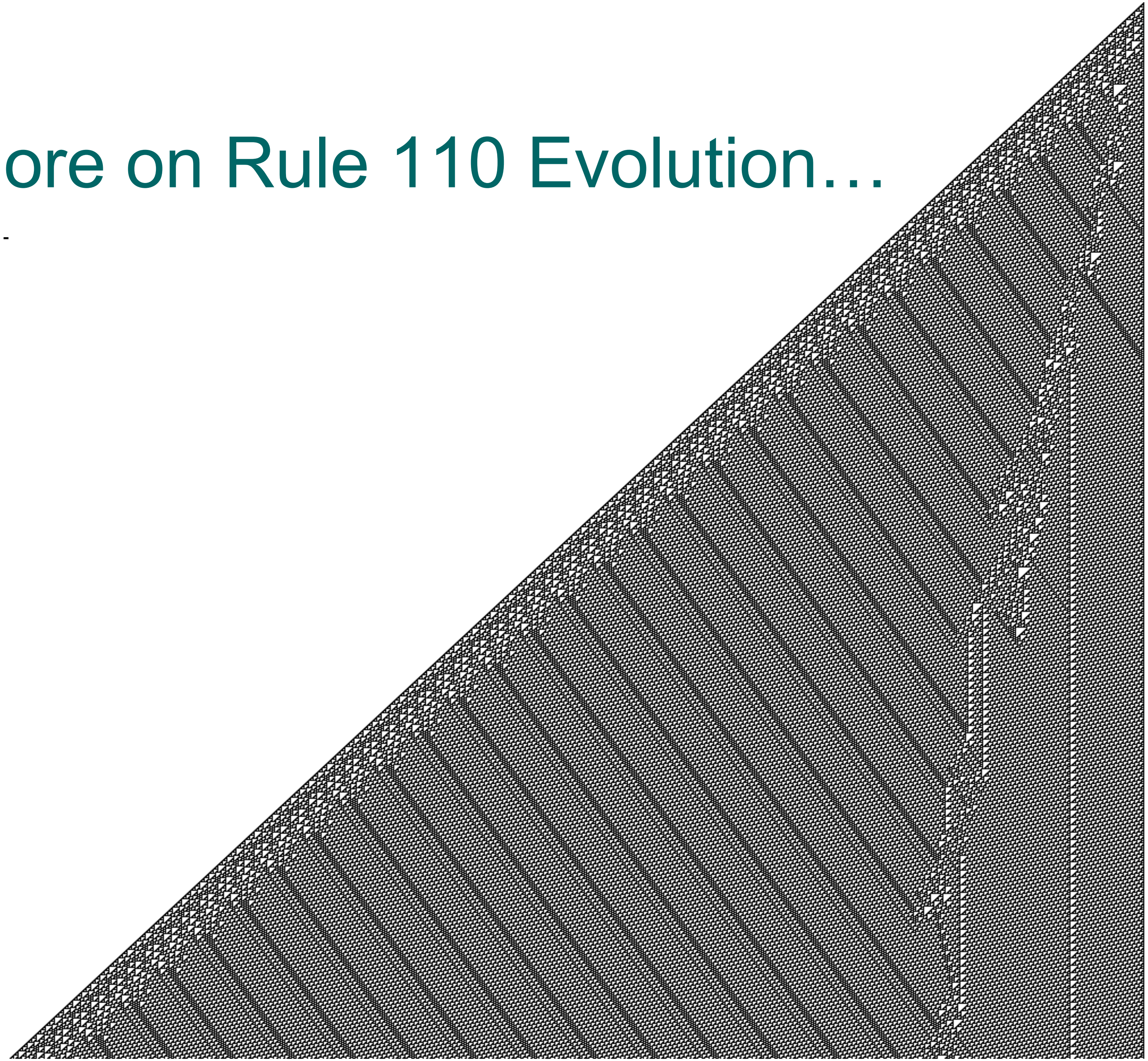


More on Rule 110 Evolution…



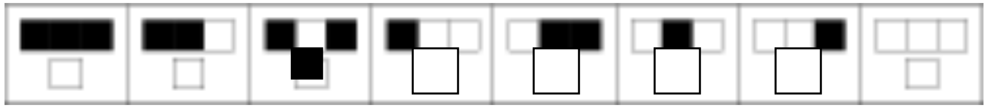
More on Rule 110 Evolution…

-



Example: Rule 32

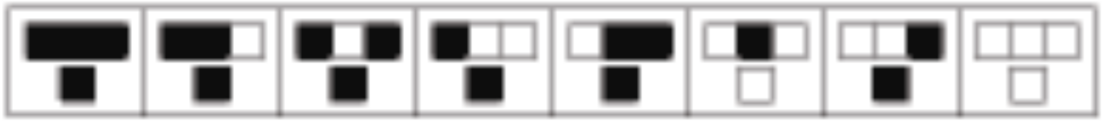
- Rules



- Steps & Evolution:
  - It is clearly that all cells rapidly becomes white

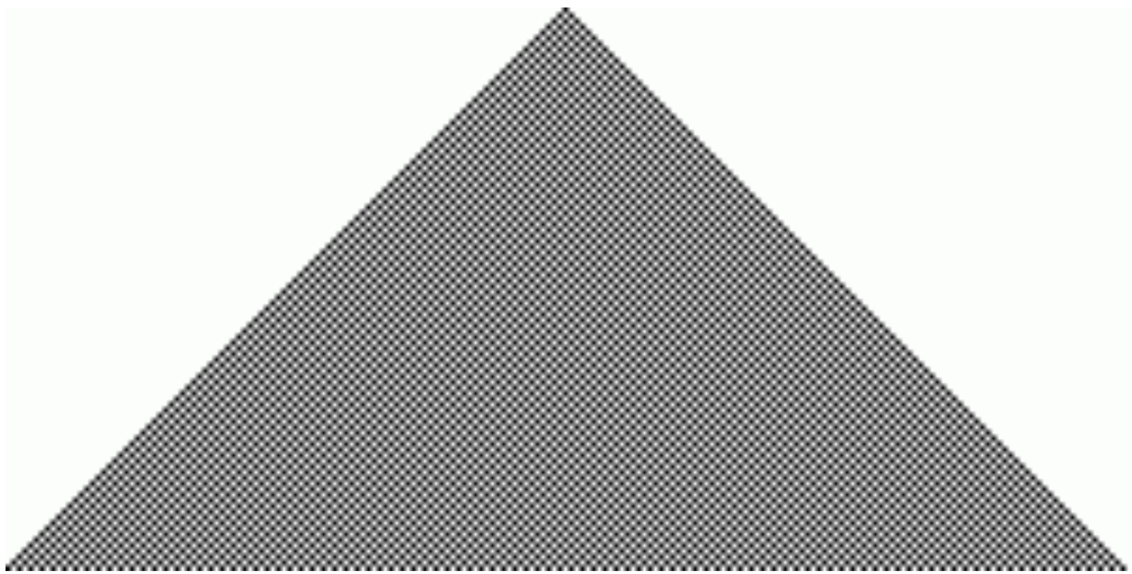
Example: Rule 250

- Rules



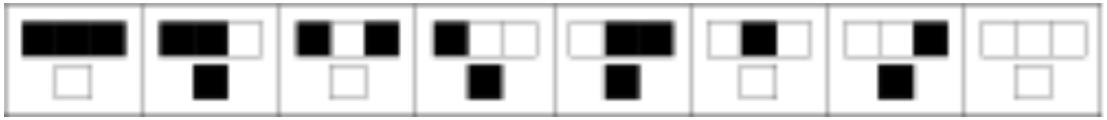
- Steps

- Evolution



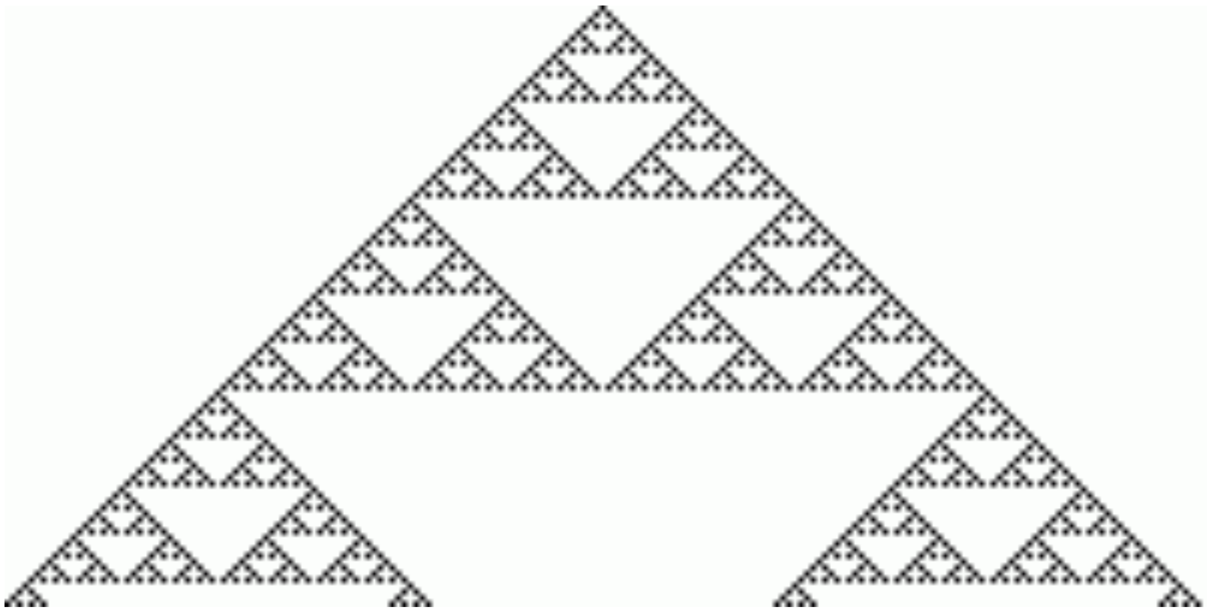
Example: Rule 90

- Rules



- Steps

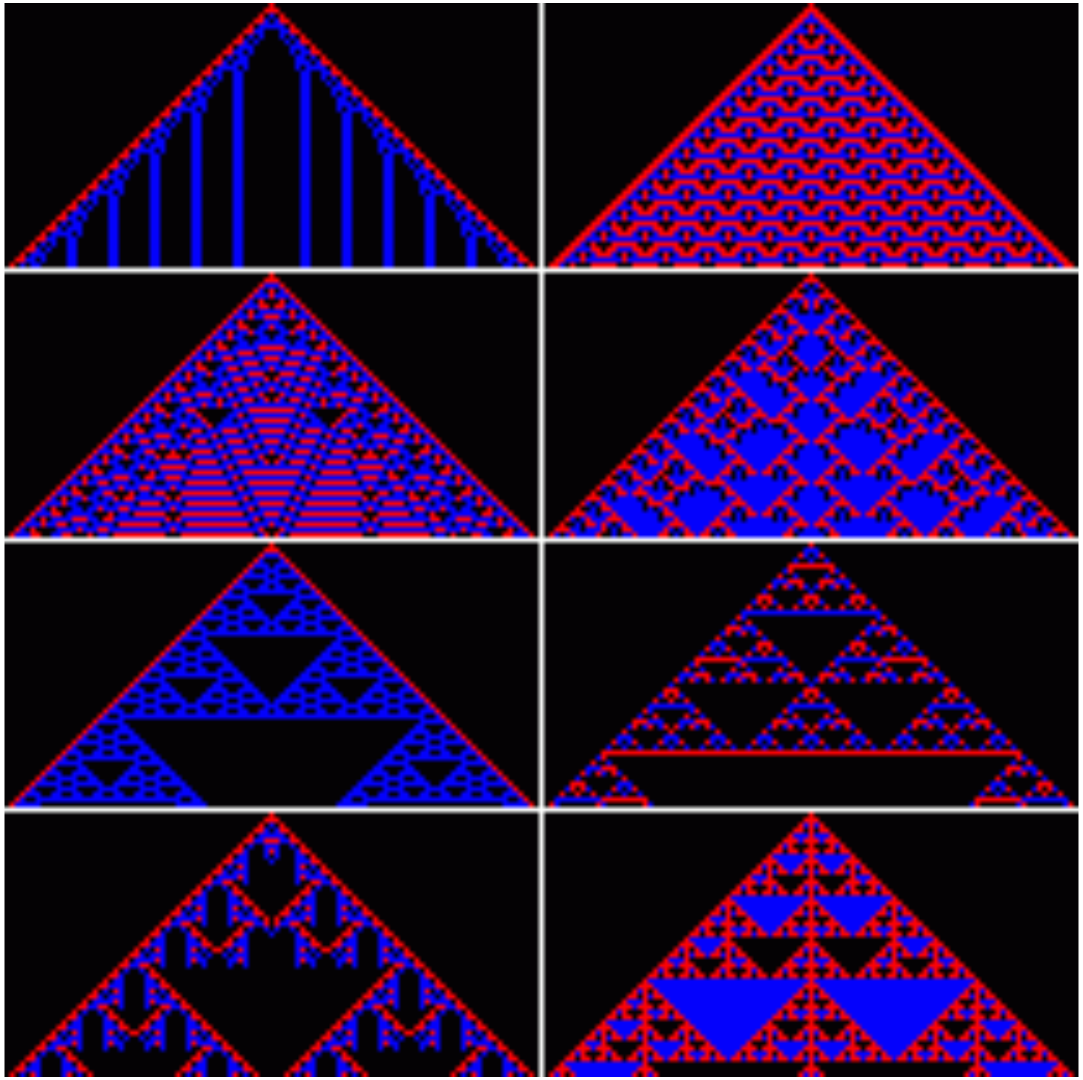
- Evolution



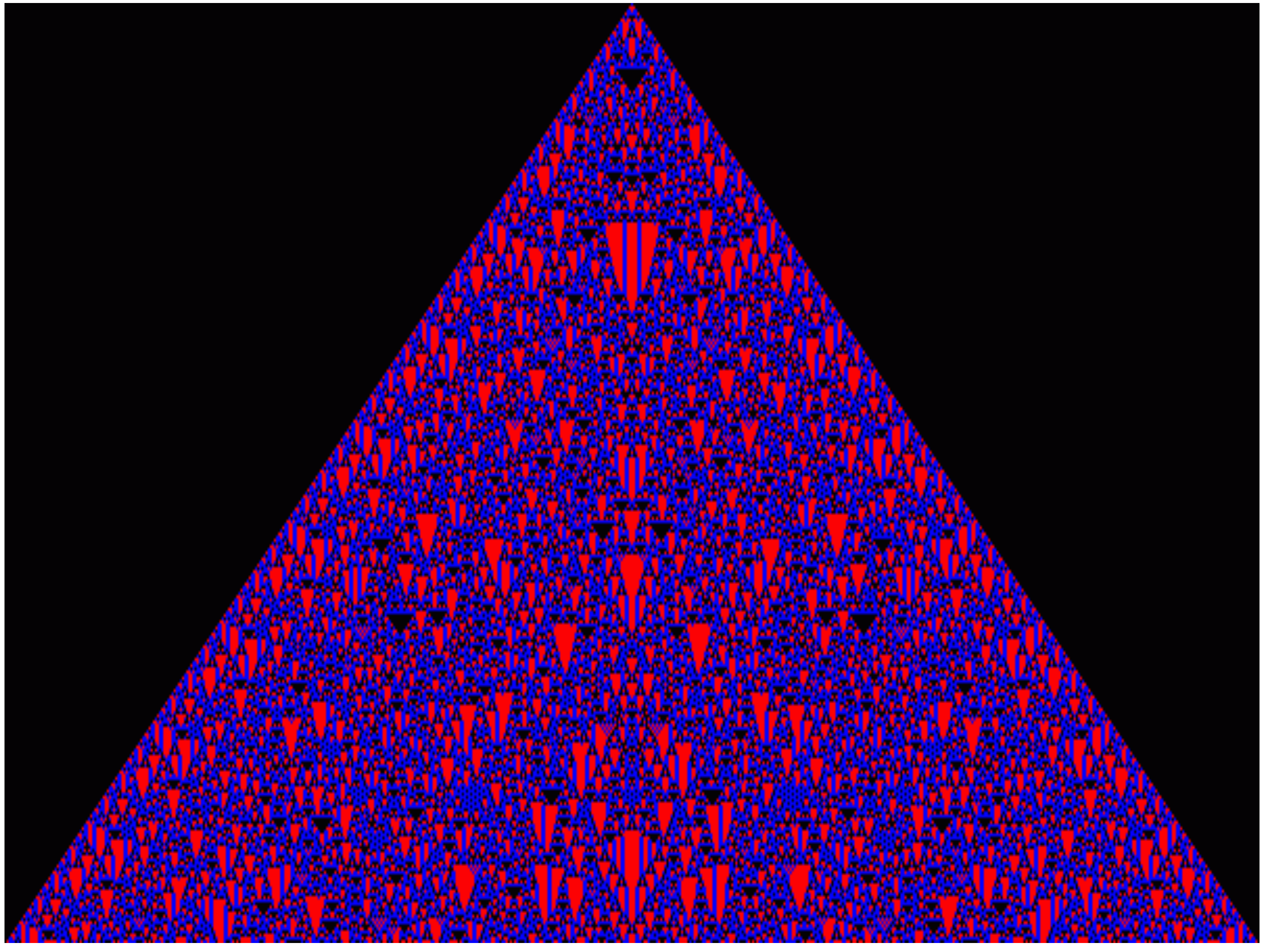
Totalistic CA

- In the rules, the state transitions in cells consider
  - The average of all neighborhood
  - Rather than the specific configurations of cells
- Let us now consider, for example
  - 1-D totalistic CA, 3-states (3 colors) with 3 neighbors

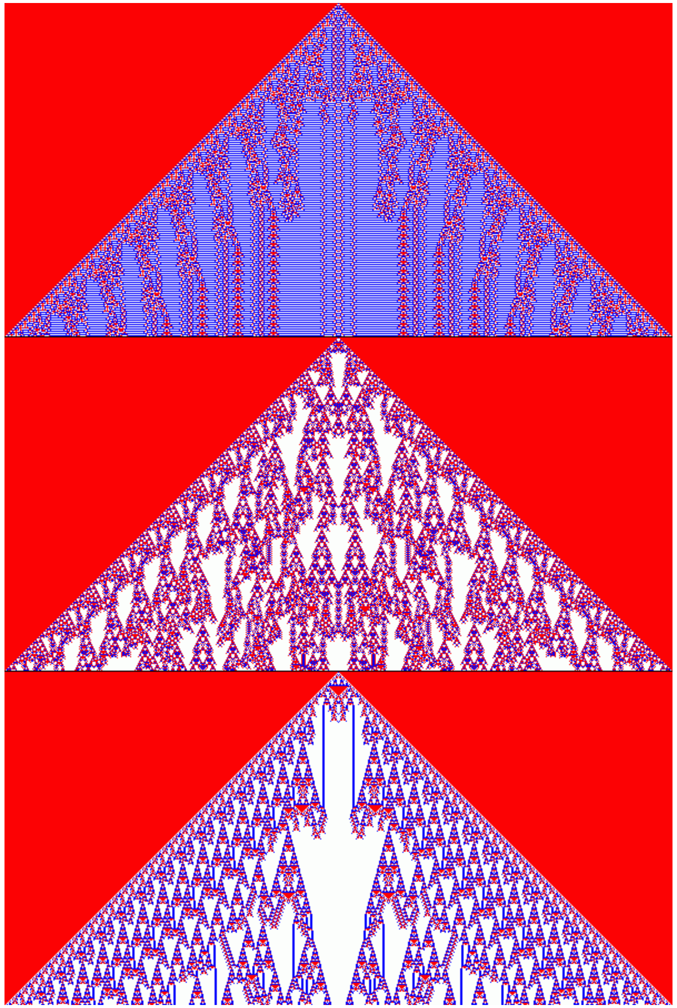
Examples of Totalistic Patterns (1)

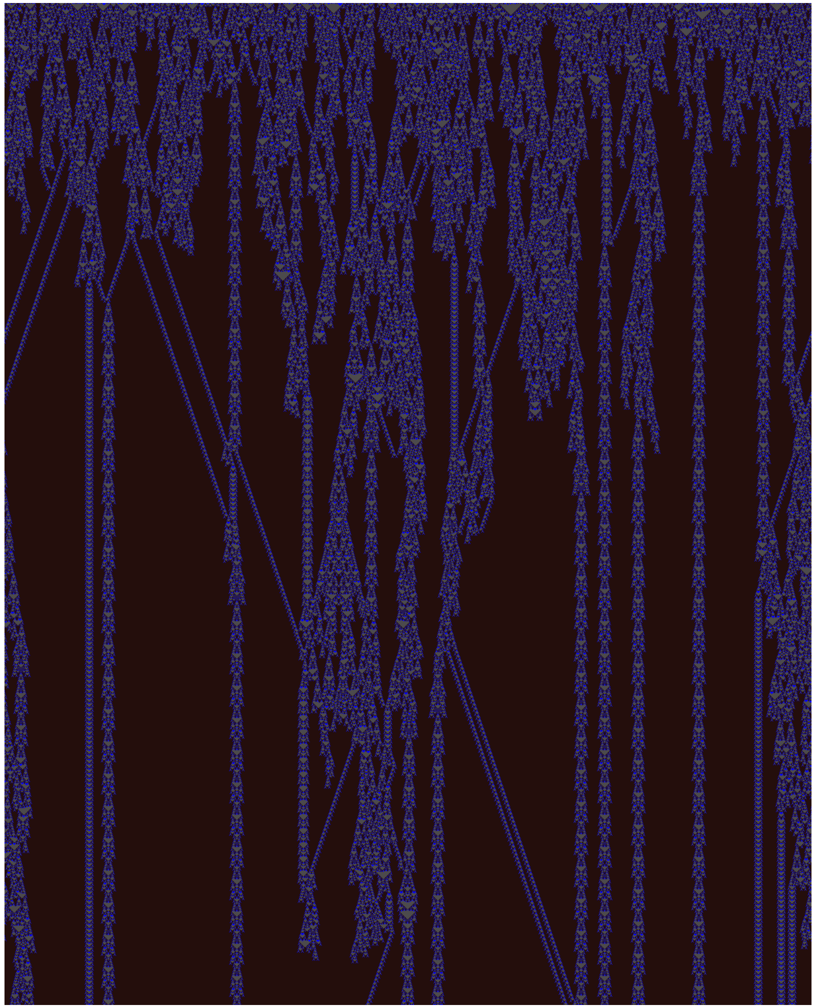


Examples of Totalistic Patterns (2)



Examples of Totalistic Patterns (3)





Part 2

- CA as Complex Systems

CA as Complex Systems

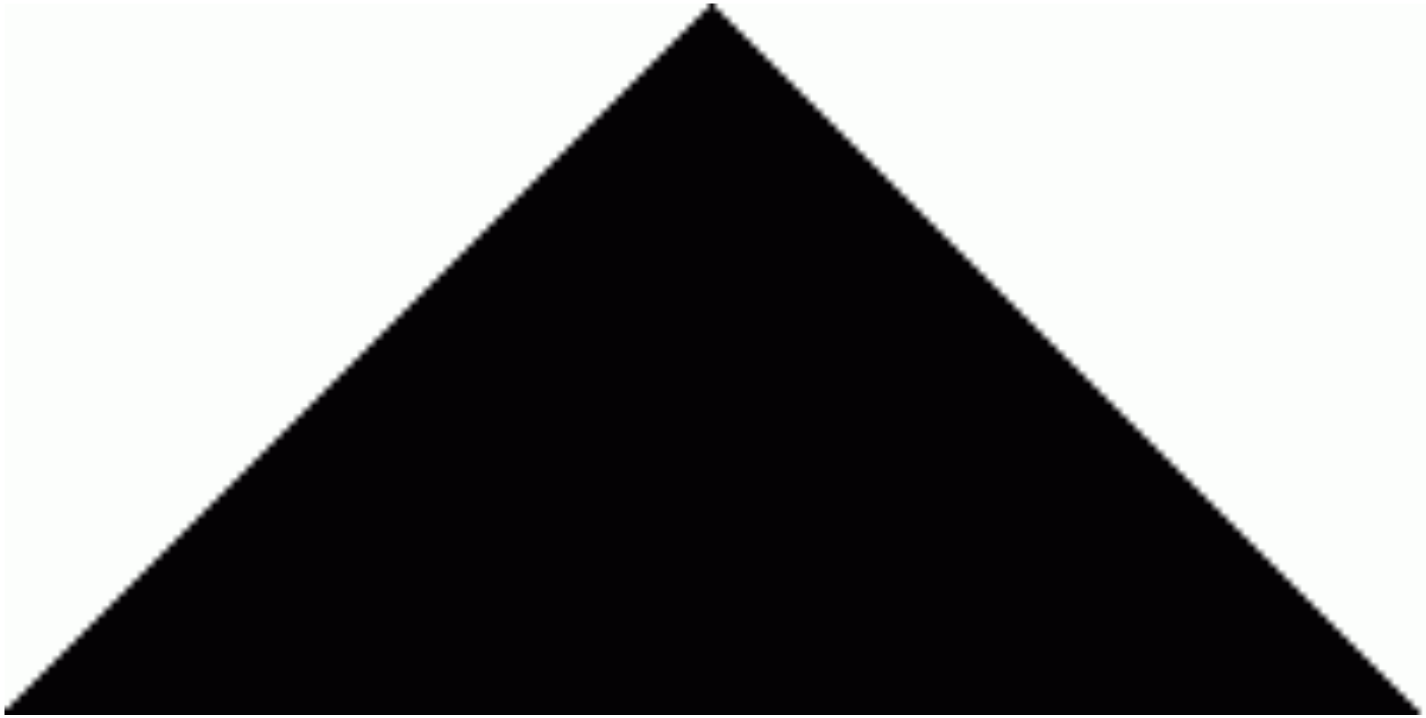
- We have seen different types of CA
  - Still, we can easily recognize that, for all types of CA, and depending on rules, the dynamic evolution of CA can fall under a limited number of classes
- The fact is:
- The possible behaviors of CA fall into a limited number of classes
  - Which are “universal” for CA
    - That is, apply to any type of CA
  - Which are “paradigmatic” of any type of complex system
    - That is, are representative of the behavior of all types of systems in nature

Classes of Behavior

- 4 Main Classes of Behavior Can be identified
- Class 1:
  - Global uniform patterns
  - E.g., all cells “white”
- Class 2:
  - Periodic, regular patterns
  - E.g., regular repetitive triangles
- Class 3:
  - Seemingly random
  - We now it is not random, but it appears like it is, no recognizable patterns
- Class 4:
  - Complex, self-organized behavior
  - Organized, recognizable structured in the middle of chaos
  - Though never really repetitive

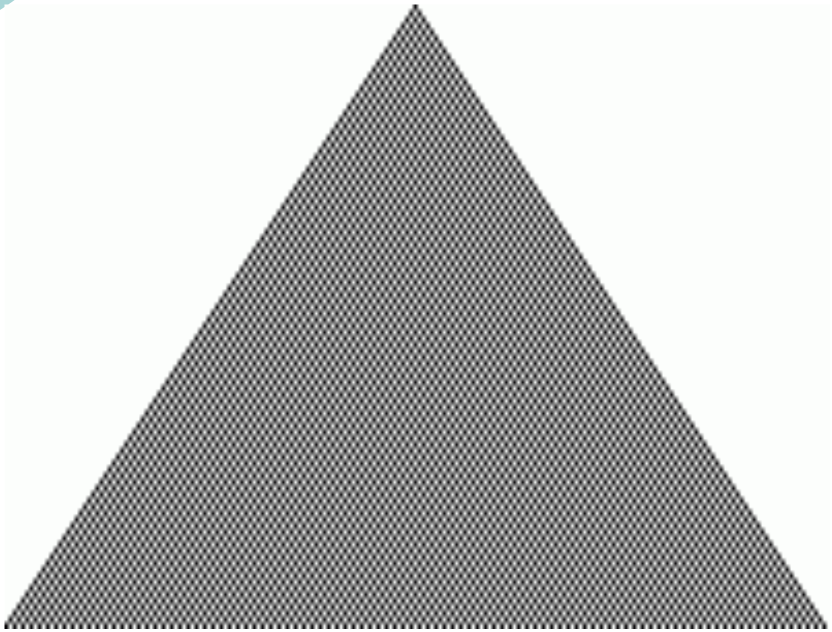
Class 1 Example

Rule 254

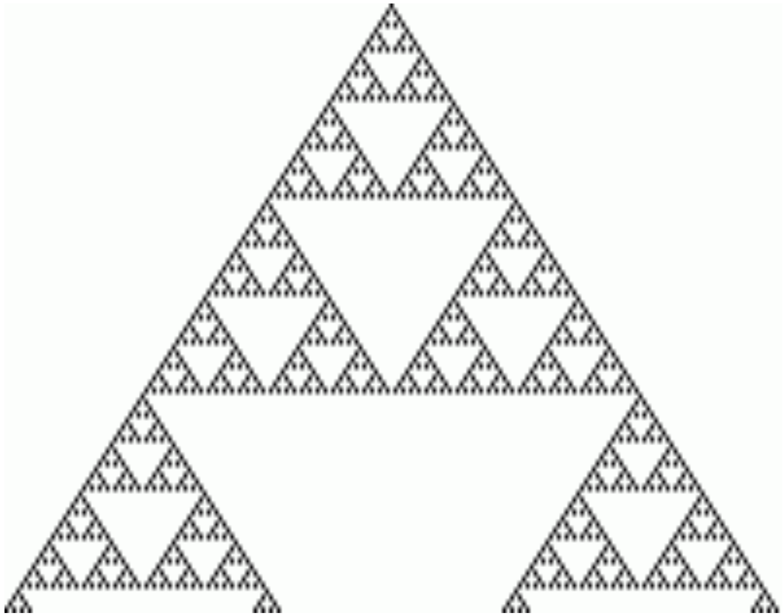


Class 2 Examples

Rule 250

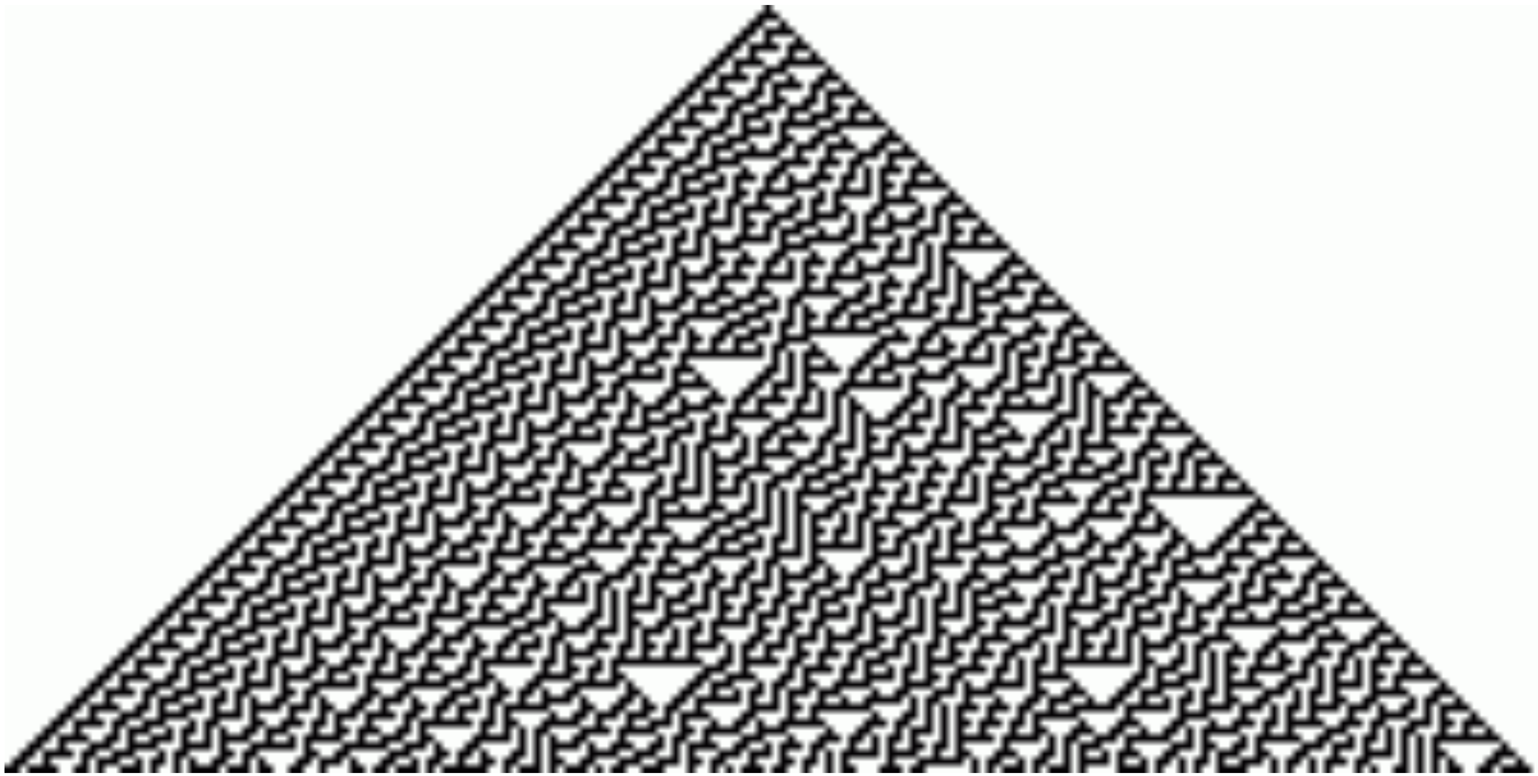


Rule 90



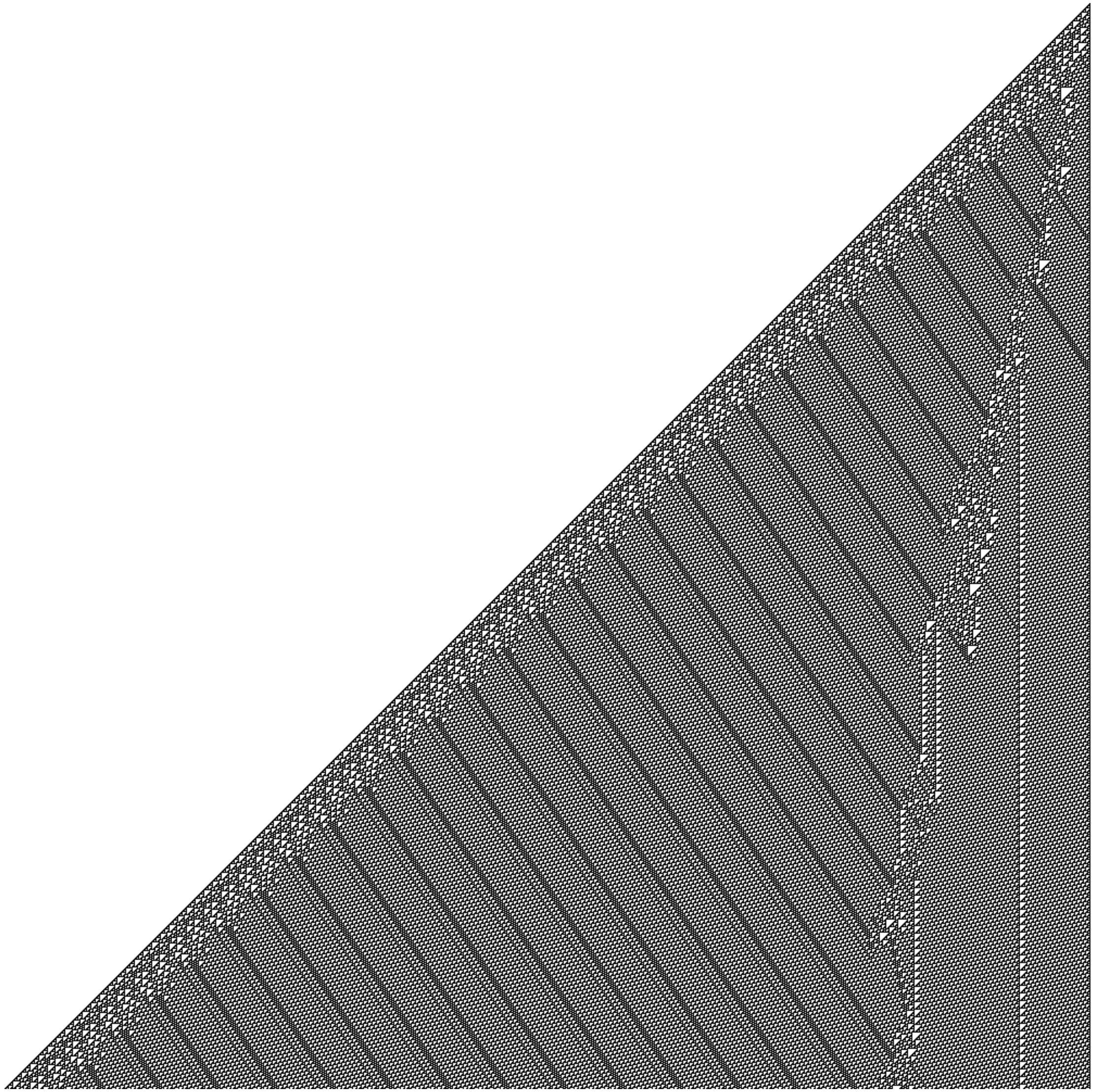
Class 3 Example

Rule 30



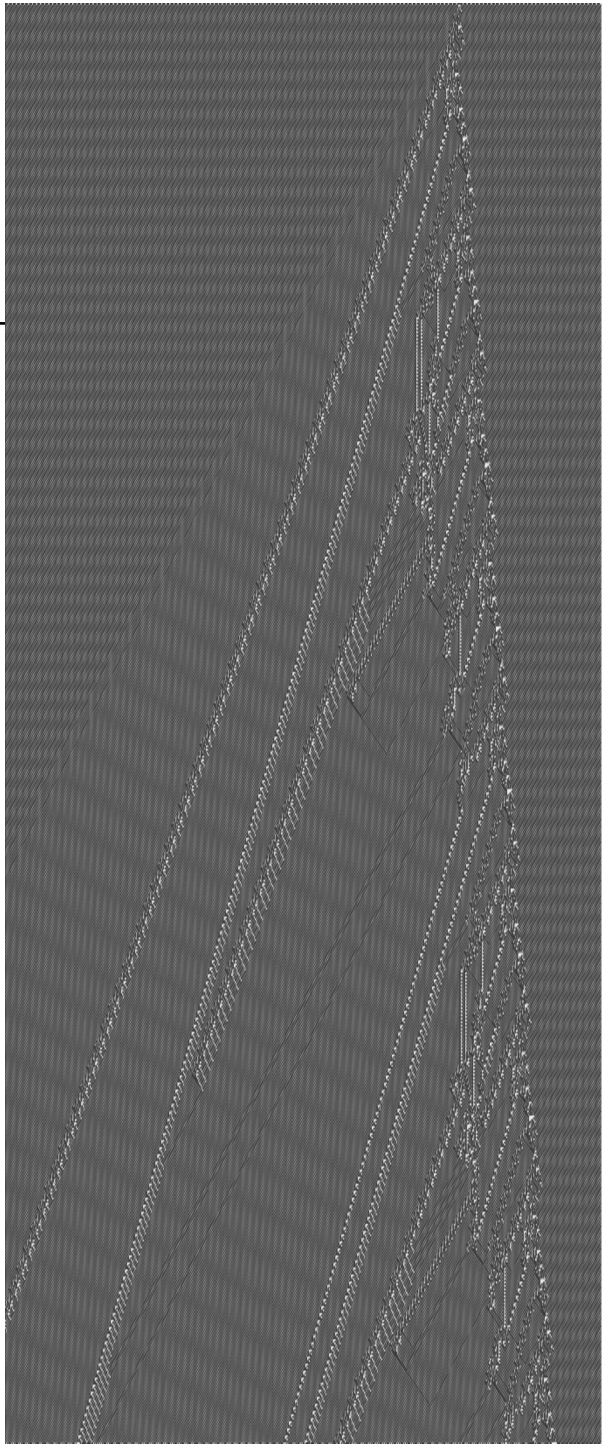
Class 4 Example (1)

Rule 110

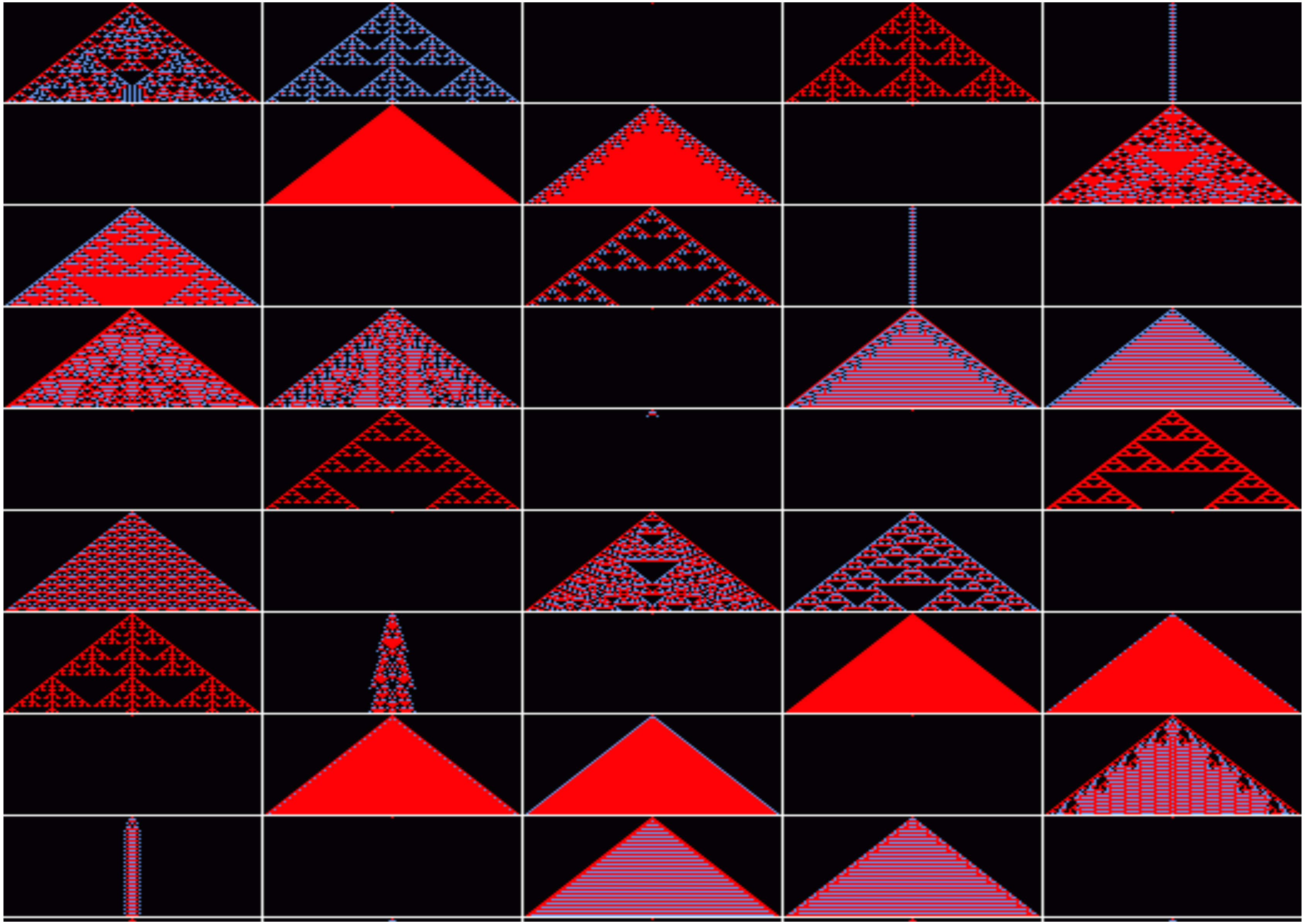


Class 4 Example (2)

Structures in Rule 110

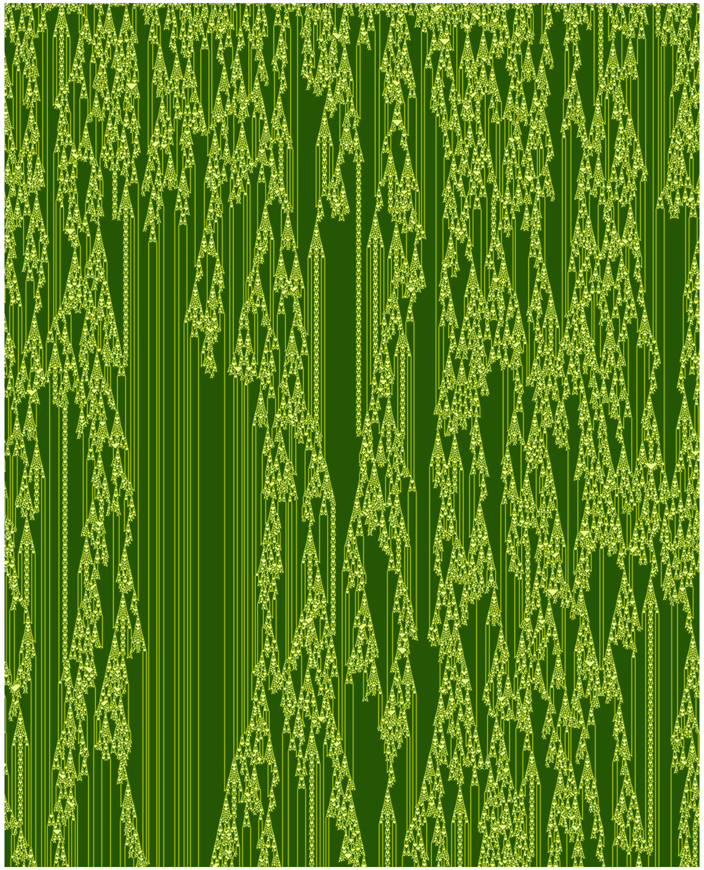


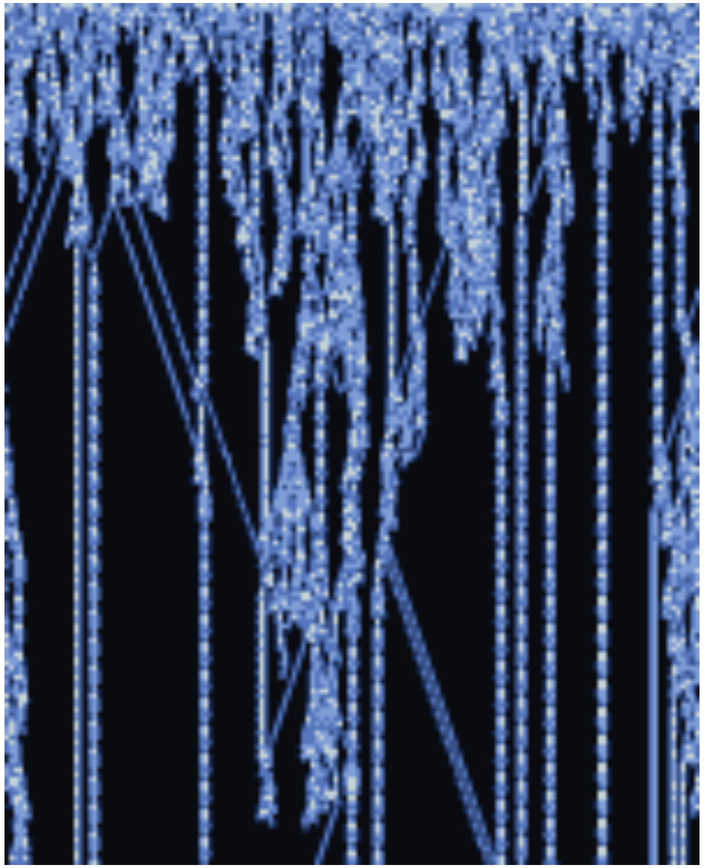
The 4 Classes in 3-color Totalistic CA



The 4 Classes in Multicolor Totalistic CA

Structures in Class 4 CA





Formalizing CA Classes

- What determines the class to which a CA belongs?
- In general, in most complex systems
  - A limited set of parameter (and often a single parameter) can be identified
  - That determines the type of dynamic behavior
- This parameter is a sort of "measure of complexity of the system"
  - And is typically denoted with "lamba" λ

The Lambda of CA

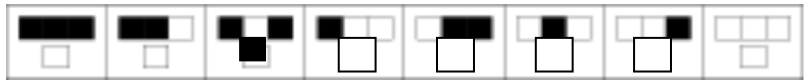
- Let us consider a simple 1-D CA with binary state
  - λ is the probability that a cell will be in state "1" in the next iteration
- It is a sort of measure of the complexity of the system
  - It measure how strong is the degree of interaction between cells
  - It measure the amount of "feedback" in the systems

CA Classes and λ

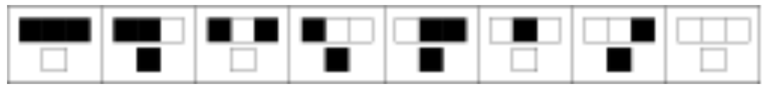
- λ close to zero
  - All cells die
  - Not enough feedback to keep the system "alive"
  - **Class 1**
- λ higher than zero and less than 0,3
  - Periodic patterns
  - Enough feedback to keep the system alive
  - Still limited enough to avoid complex interactions
  - **Class 2**
- λ closer to 0,3
  - Stable not-periodic structures emerge
  - The interaction feedback is quite strong, but it still preserve the possibility for regular self-organized structures to exists
  - **Class 3**
- λ high than 0,3
  - Random, chaotic behavior
  - To much feedback "noise" to sustain any regular behavior
  - **Class 4**

CA Classes in the Elementary CA

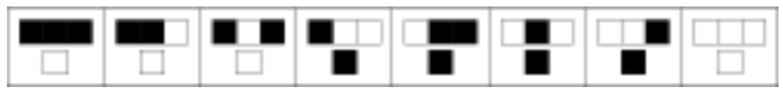
- Rule 32: λ=0,03



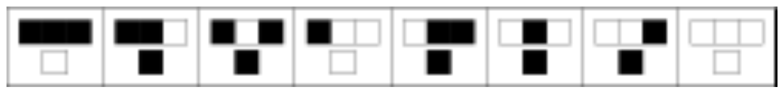
- Rule 90: λ=0,2



- Rule 110: λ=0,3

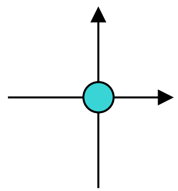


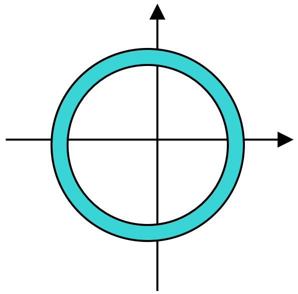
- Rule 30: λ=0,5

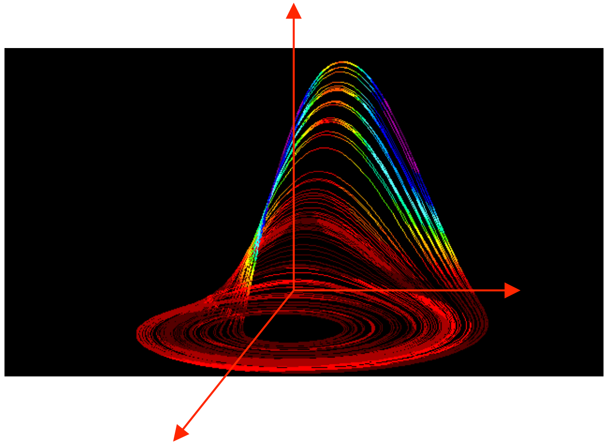


CA and Attractors

- Reasoning with reference to some sort of "phase space" for CA
  - Class 1: a single limit point attractor
    - The system stabilize in a static way
  - Class 2: one or several limit cycles
    - The system stabilize in a periodic dynamic pattern
  - Class 3: complex attractors
    - The systems evolves in an attractor which is a complex manifold, but which still ensures some sort of "bounded" behavior
  - Class 4: chaotic attractor
    - The system evolves in a very chaotic manifold, with not regularity





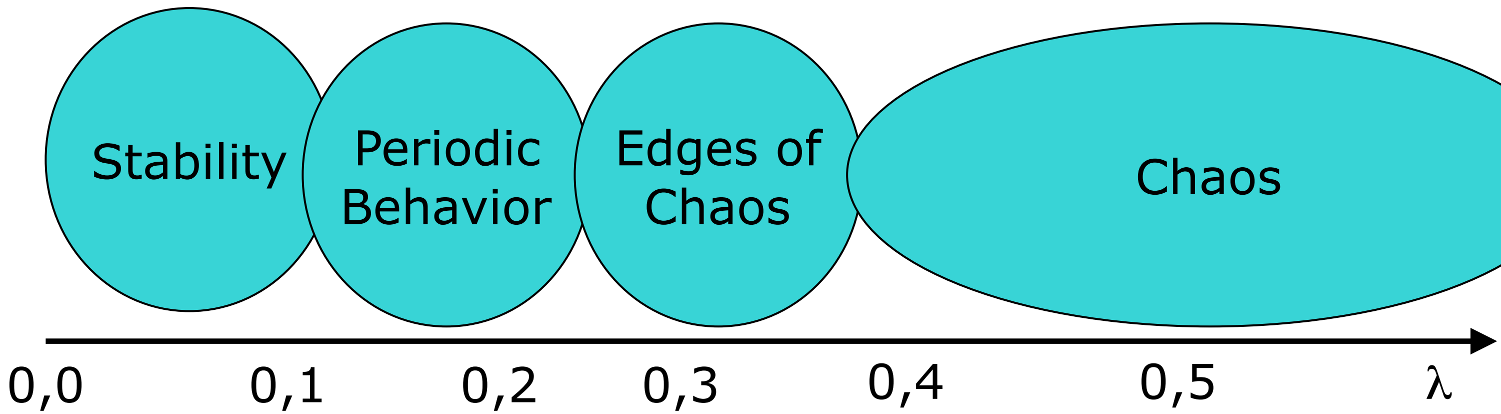


Universality of CA

- All types of CA, defined a proper λ, exhibit very similar behavior
  - A complex CA does not lead to more complex behavior
  - A behavior is always in one of the 4 classes
- All types of complex systems exhibit similar characteristics
  - Some sort of λ exists such that
  - Its tuning move the system from simple, to periodic, to complex, to chaotic behavior

The Edges of Chaos

- For any type of complex system, we can identify “regions” of behavior in function of some relevant parameters (e.g., λ)
  - The shift from order to chaos involves a region in which complex behavior emerges
  - The Edge of Chaos



Adaptive Self-organization and the Edges of Chaos

- At "the edges of chaos" a system exhibit complex patterns
  - That although never repetitive
  - Shows some regularity
  - Have spontaneously emerged
  - There is some sort of "self-organization"
- The patterns live and survive in a continuously perturbed world
  - Neighbor cells continuously interact with the pattern
  - And the pattern nevertheless survice, possibly changing its shape
  - ADAPTIVITY!!!
- Actually, most interesting natural systems are at the "edges of chaos"
  - The heart
  - Living organisms
  - The Internet and Complex Networks
  - Will see in the following of the course…

Part 3

- Asynchronous and Stochastic CA

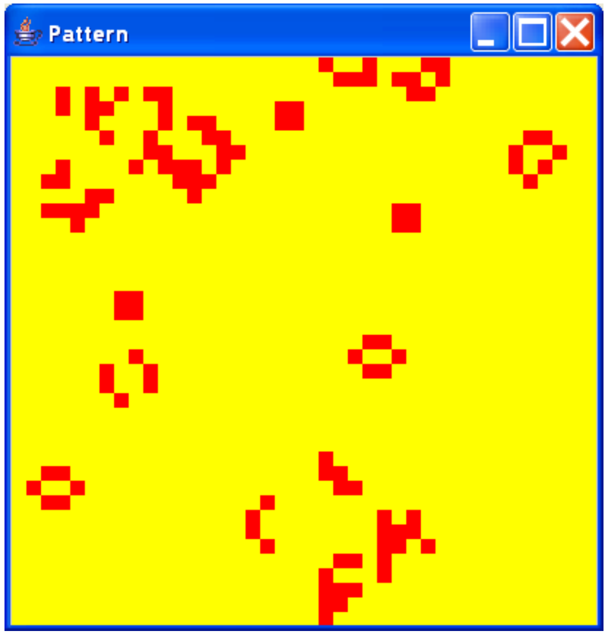
What are Asynchronous CA

- Asynchronous CA, unlike Synchronous one
  - Are not globally synchronized for state transitions
- Each cell evaluate the state transition function independently of other cells
  - And then change its state immediately
- When a cell change its state?
  - When commanded by an external entity
    - Not really an interesting case
  - Autonomously, based on some internal autonomous decision making
    - Randomly from time to time
    - Based on an internal clock
- Asynchronous CA provides for a more continuous notion of time
  - The time is no longer a sequence of time frame
  - But it is a ***continuum***, in which cells acts independently of each other

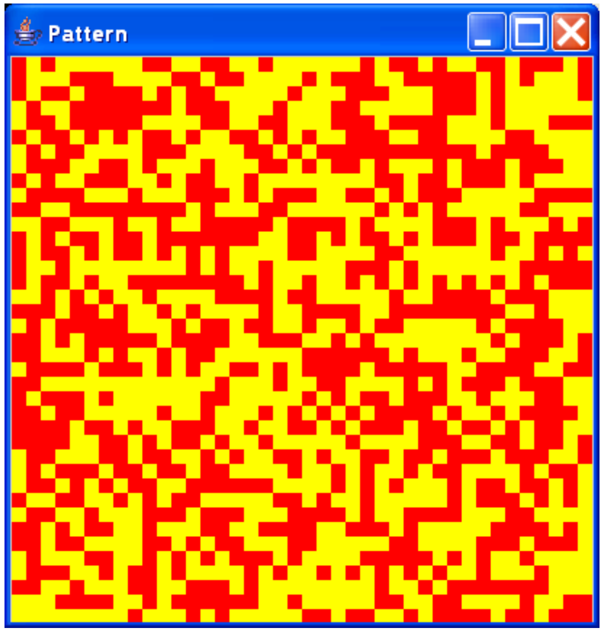
Dynamics of Asynchronous CA (1)

- The Behavior of Synchronous CA is dramatically different from that of Asynch
- Consider, e.g., the "Game of Life" rules

Synchronous Evolution



Asynchronous Evolution



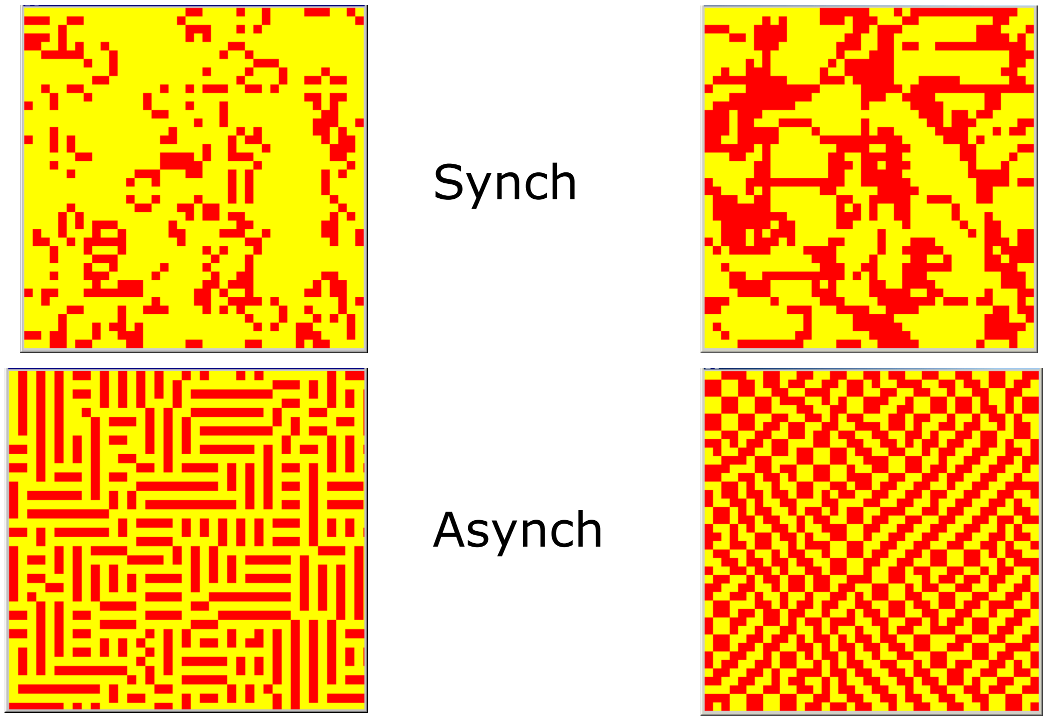
Dynamics of Asynchronous CA (2)

- In general
  - A final state that is stable for a synchronous CA is stable also for an asynchronous CA
- The attractors are the same
- However
  - The evolution path of an asynchronous CA is much different
  - Tends to reach different final configurations
- First, it has a natural endeavor to self-organize itself
  - Some sorts of “embryonic” patterns emerge
  - For any rule
- And tends to be “stable”
  - The evolution stabilize very rapidly
  - While synchronous CA tend to fall into cyclic attractors

(b)

Dynamics of Asynchronous CA (3)

- Other examples of differences between synchronous CA and their asynchronous counterparts



Asynchronous CA and Complexity

- Asynchronous CA somewhat appears to be more realistics
  - Ecological systems are asynchronous (animals act based on internal clocks)
  - Social systems are asynchronous
  - Distributed systems are asynchronous
- At the same time, they exhibit
- more stable behavior than their synchronous counterparts
- A natural facility in exhibiting simple self-organizing patterns
- But where does complexity and adaptivity comes from in the real-world ????

Stochastic Cellular Automata (I prefer to call them "Perturbed CA")

- Consider a normal Cellular automata

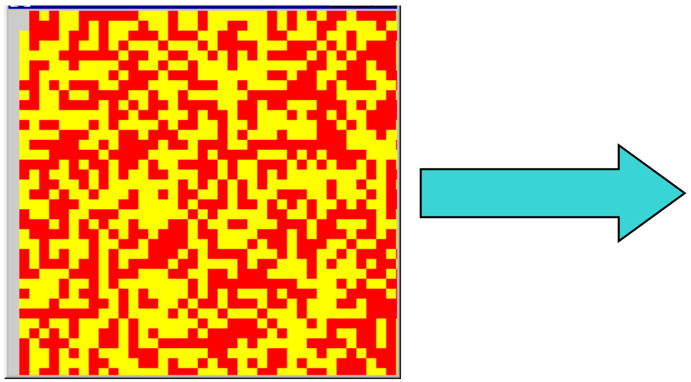
  $A=(S,d,V,f)$

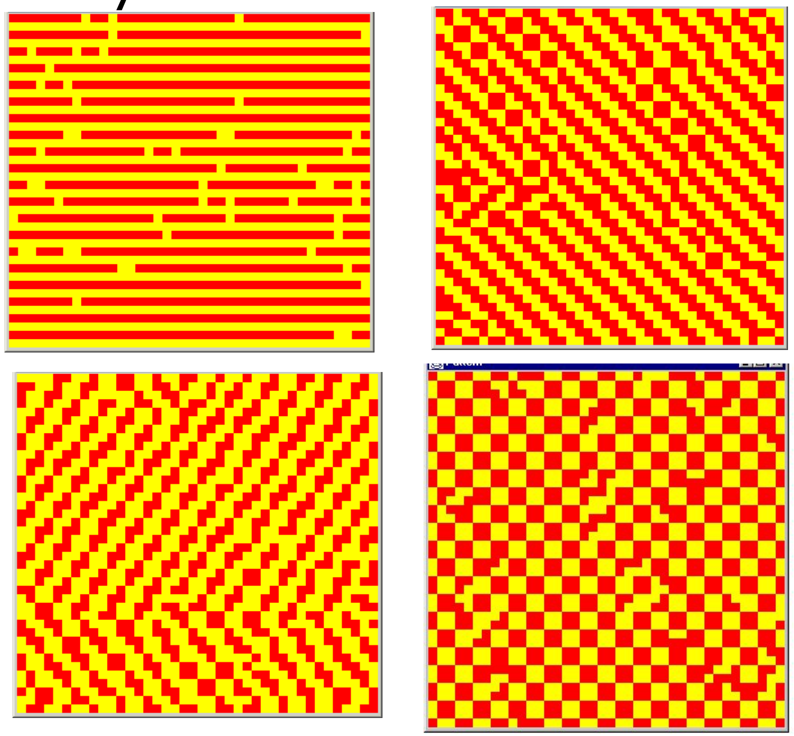
- But asynchronous with
  - Cell are autonomous
  - time continuous dynamic

- And in which some stochastic phenomena overrules normal rules for state transitions
  - As if cells were "not rational" OR $\varphi(\alpha, D)$
  - As if cells were situated in an environment in which external **perturbation** can change artibrarily the state of cells

Emergent Behaviors in Perturbed CA

- Global scale self-organized behaviors emerge from even simple CA, similar to the ones observed in real world systems



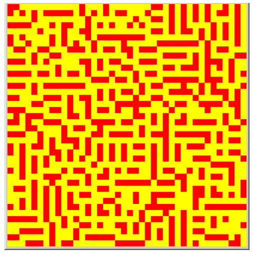


When and Why?

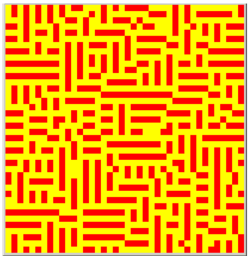
Internal vs. External Dynamics in Perturbed CA

- Let us consider the relative dynamics of
- The states changing their internal state
  - Probability frequency λ_i
- The environment forcing some change of state in a cel
  - Probability frequency λ_e
- And evaluate what happens depending on the λ_i/λ_e rati
  - Which simply characterize the “strength” of the perturbation

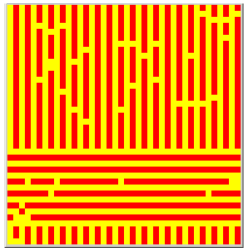
Impact of Perturbation Dynamics



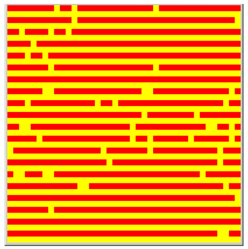
$\lambda i/\lambda e = 0.001$



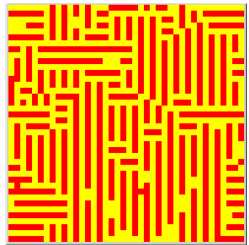
$\lambda i/\lambda e = 0.01$



$\lambda i/\lambda e = 0.02$

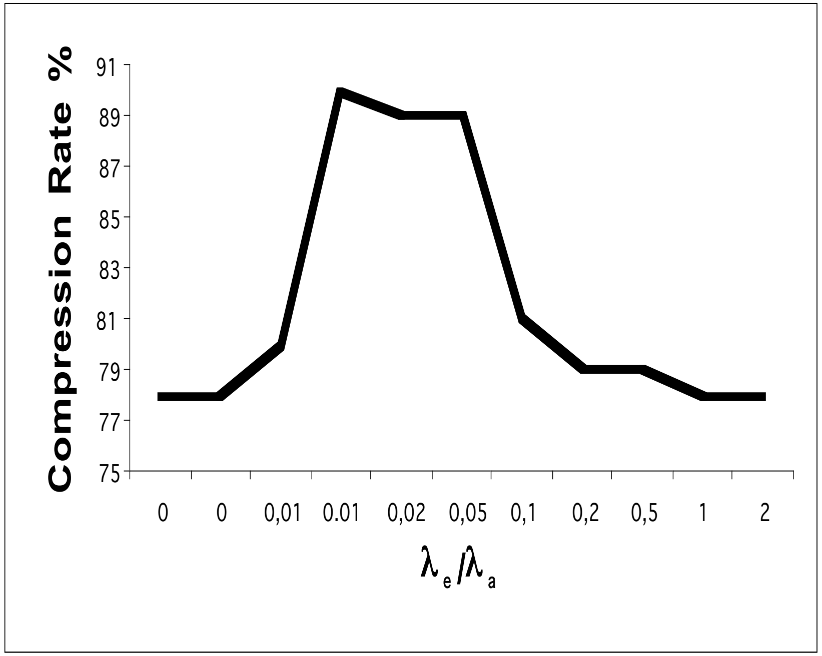


$\lambda i/\lambda e = 0.05$



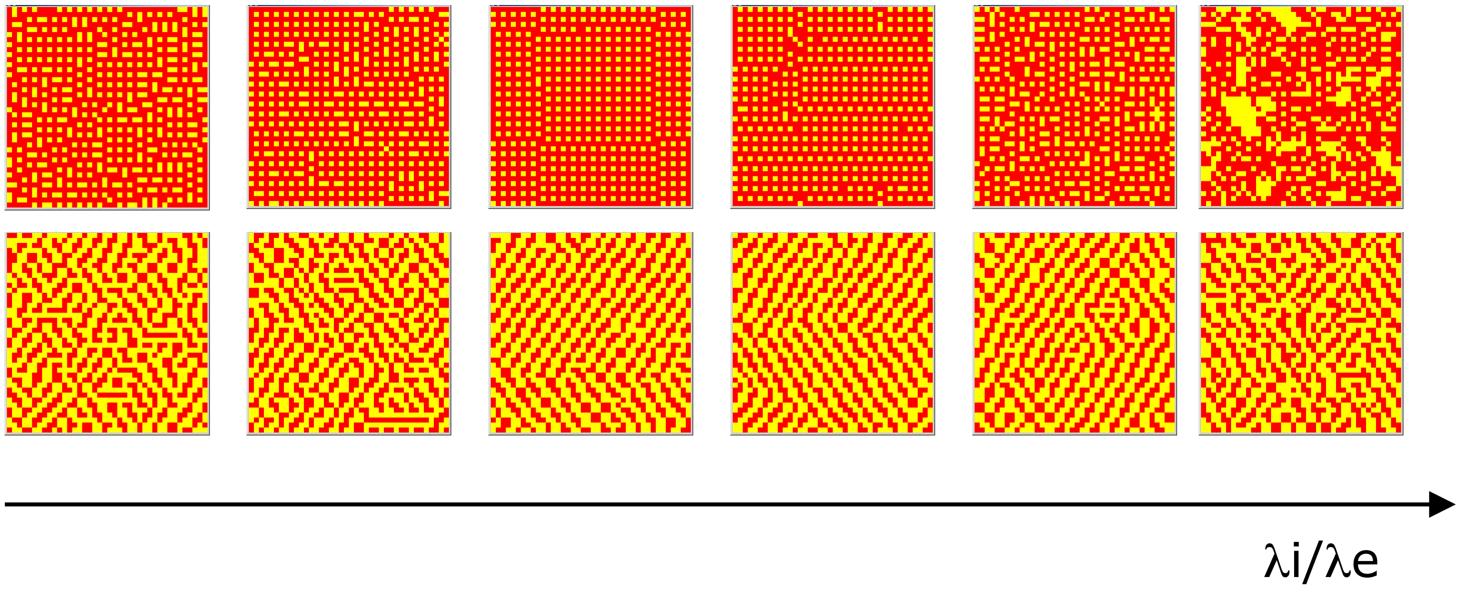
$\lambda i/\lambda e = 0.1$

- Low perturbation dynamics
  - No global structures emerge → quasi equilibrium
- Moderate perturbation dynamics
  - Global structures → spatial patterns
- High perturbation dynamics
  - Turbulence! → Chaos!



Generality of the Phenomena

- The same phenomena occur for a large variety (nearly all) of asynchronous CA types
- And the same occurs in 1-D CA



Implications of the Phenomena (1)

- Real-world system tends to “locally organize”
  - And to stabilize into local self-organizing patterns
  - These may be somewhat “complex” but have no global impact
  - Local minima of equilibrium
- In the presence of environmental dynamics
  - Perturbations always move local configurations out of equilibrium
- Then they tend to re-stabilize but
  - During evolution, only more and more global equilibrium situations survive
  - Until self-organization patterns becomes global
  - In a sort of continuous “simulated annealing”
- Overall, emergence of global spatial patterns from
  - Local interactions
  - External perturbations

Implications of the Phenomena (2)

- The emergence of global spatial patterns in perturbed CA is the phenomena that occur in:
  - Sand dunes, granular media, population distribution, etc.
- So, we have eventually completed the pictures on CA as complex systems,
  - Showing the impact of the last ingredient of the “complexity” recipe (openness and environmental dynamics)

Part 4

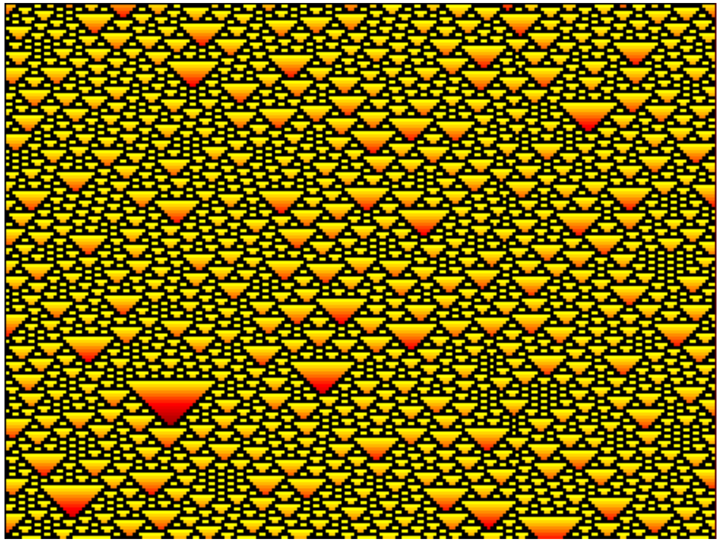
- Applications of CA

What are CA useful for?

- For the study of complex systems
  - We have already seen in the previous part
  - CA as paradigmatic of more complex complex systems
- For the understanding and simulation of complex natural behavior
  - Chemical, physical, ecological, biological
- For the study of complex distributed systems
  - Multiagent systems
  - Complex networks
  - Information and virus propagation

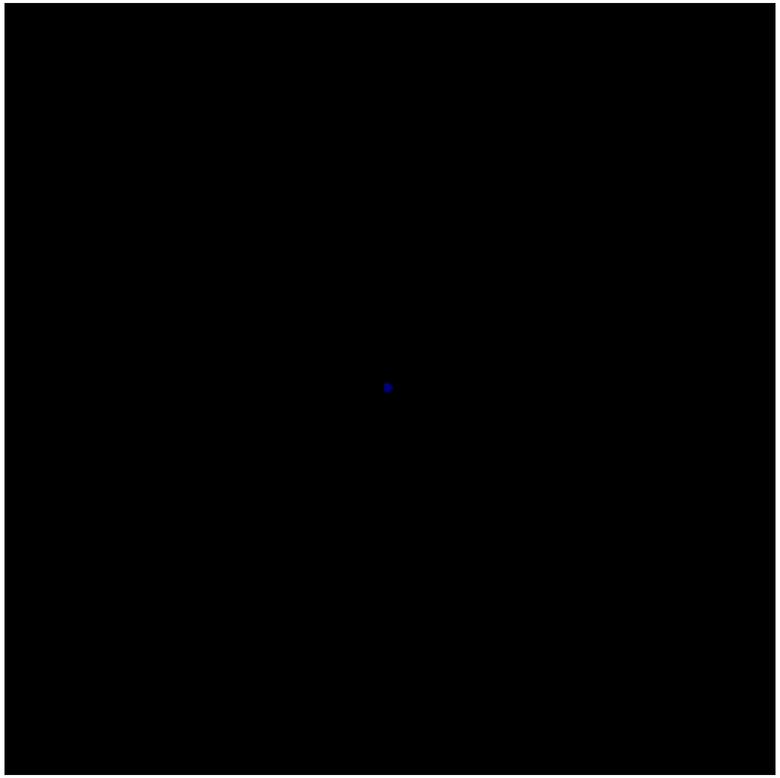
Phenomena of Growth: Cone Shell Patterns

- The shell grows up linearly at its borders
  - Pigmentation at a specific point is influenced by the pigmentation around the point of growth
- It's a 1-D CA?
  - Yes, to some extent
  - Although pigmentation is not really discrete, by subject to chemical diffusion
- Reaction-diffusion model is better
  - Will see in the future



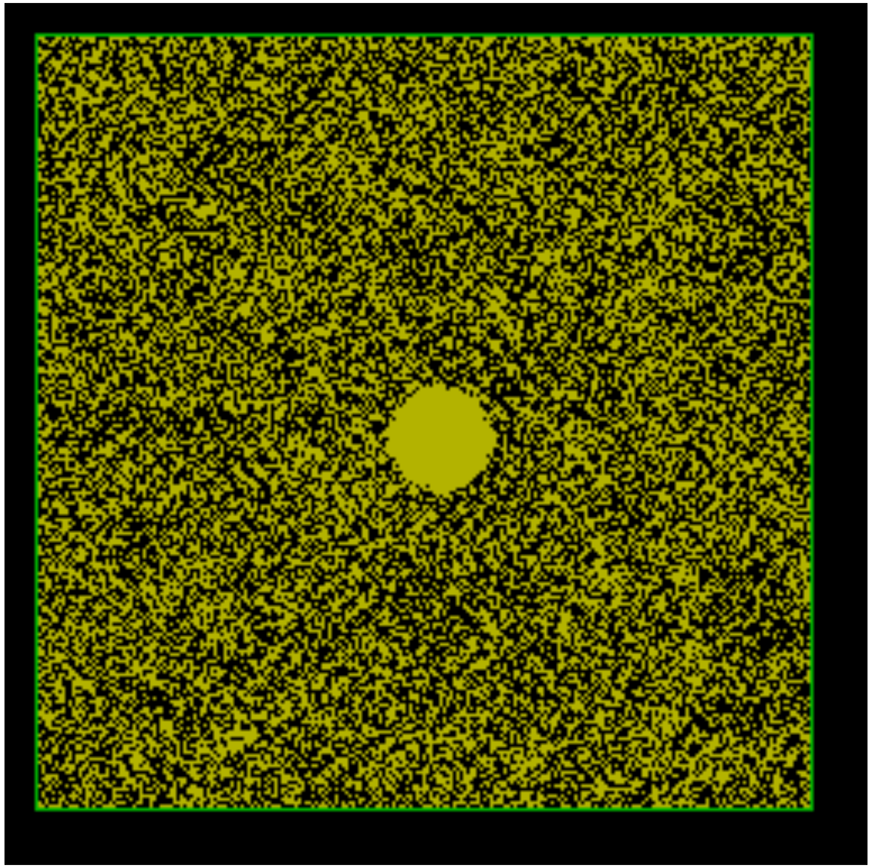
Phenomena of Growth: Snowflakes

- New ice crystals attach to the existing structure
  - With simple “CA-like” rules
  - On a hexagonal lattice



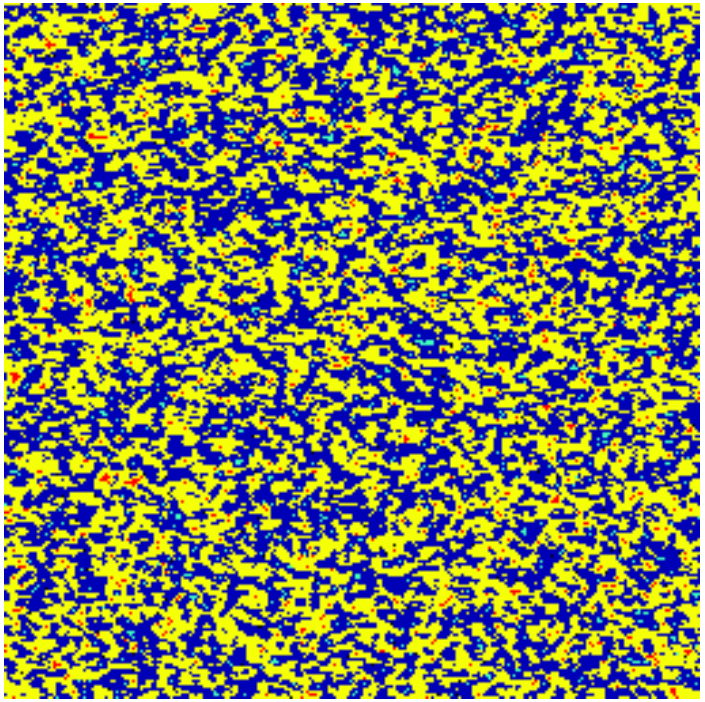
Physical Phenomena: Wave diffusion

- Waves propagates in a simple way
  - Via local interactions among particles
- Can be simulated via a simple CA
  - The state of a local cell representing the pressure



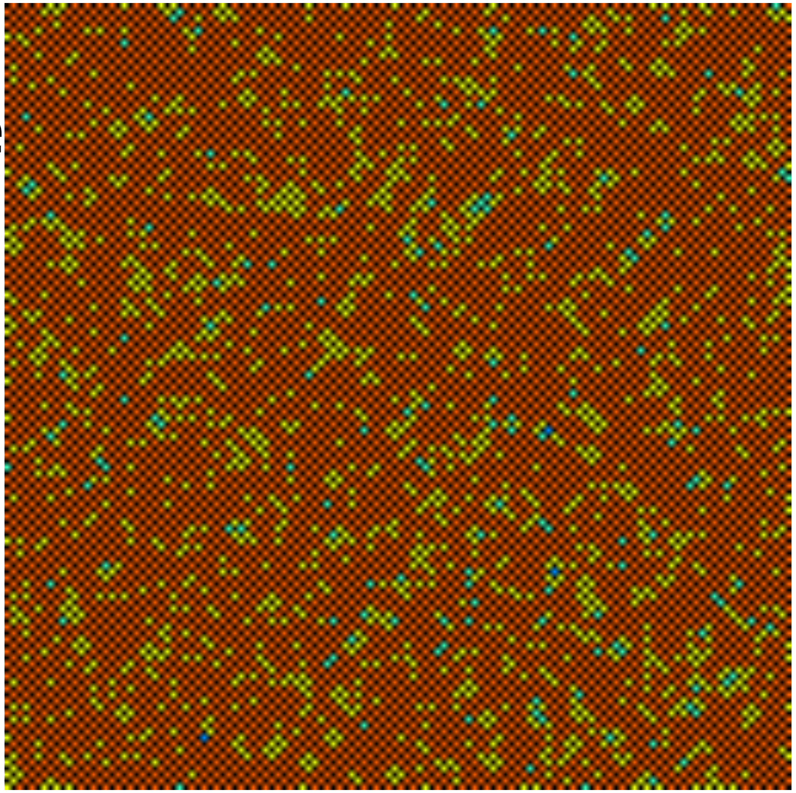
Chemical Phenomena: Simulating a Reaction

- Chemical reactions take place via local interaction of chemical components in a space
- Can be simulated via 2-D CA
  - The state of a CA represent the concentration of components in a point
  - Transition rules determines how concentrations changes during the reactions



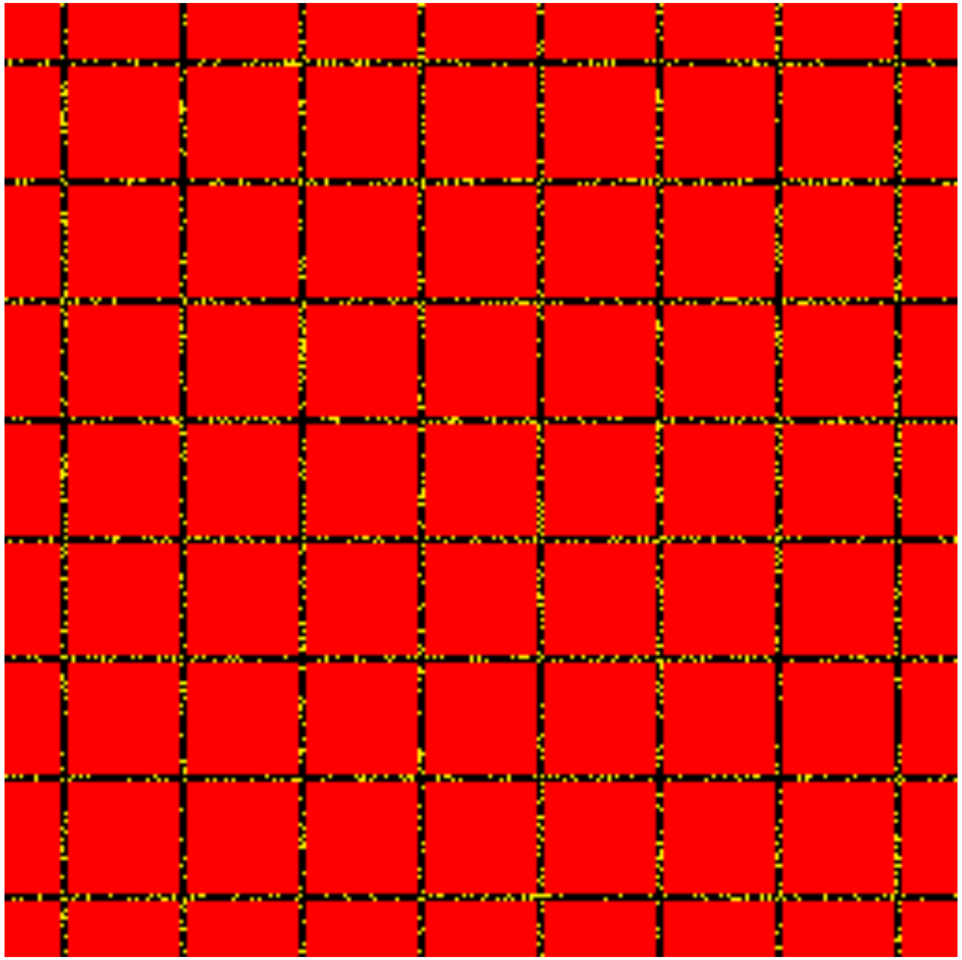
Ecological Simulation: Forest in Fire

- The state of the cell determines the presence of tree and of fire in the forest
- Rules specify how fire propagates



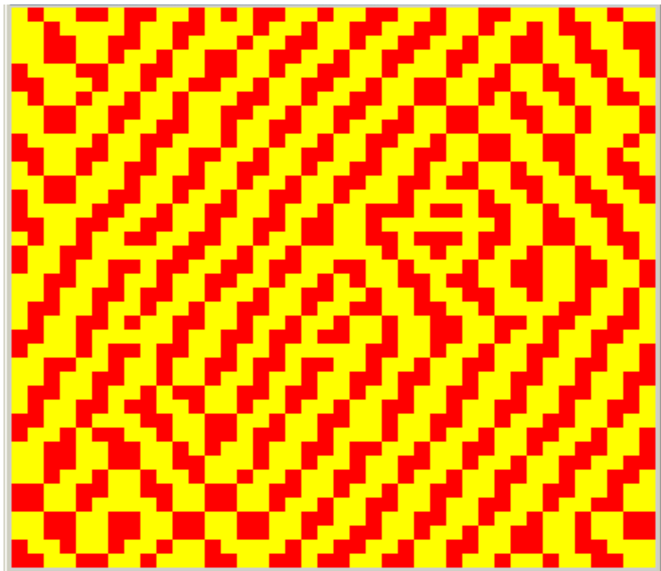
Social Phenomena: Traffic Simulation

- Traffic Simulation
  - The state of cells determines the presence of vehicles
- Rule transitions determines how a vechicle can move
  - i.e., how a state can be transferred to a close cells



Perturbed CA and Patterning of Animals

- Strip-Like Patterning of Animals
  - Cells grows and pigments diffuse in a continuously perturbed environment
  - The results is a sort of globally organized pattern



Perturbed Asynchronous CA and Modern Distributed Systems (1)

- Asynchronous CA resembles network of agents
  - Autonomous components (their actions, i.e., state transitions, are not subject to a global flow of control)
  - Interacting with each other in a local way
  - As multiagent systems and, more generally, modern distributed & decentralized computing systems
- And they are situated too
  - Their actions consider and are influences by the environment in which they situate
  - Influencing state transitions

- So, asynchronous perturbed CA are a sort of "minimalist" modern distributed system

Perturbed Asynchronous CA and Modern Distributed Systems (2)

- Given the similarities
- It is likely that the same types of globally self-organized behaviors emerge in large open distributed systems
  - Global patterns of coordination of activities
  - Global patterns of information flow
- This can be ***dangerous***
  - Unpredictable "resonance" like behavior, depending on the environmental dynamics
  - E.g., global price imbalances in computational markets, global imbalances in resource exploitations, biased sensing activities in sensor networks
- But it can be also ***advantageous***
  - E.g., global (indirect) coordination achieved by "injecting" energy via the the environment
  - A very simple and effective way to achieve global scale coordination and/or global scale diffusion of information

Conclusions and Open Issues

- CA are a very important class of complex systems
  - Representative of a larger class of systems
  - Useful for simulation
  - With possible implications for modern distributed systems
- But there are issues that CA are not able to consider
  - CA use only regular grid-like networks
    - What is the influence of the network structure?
    - What is the influence of non-local network connections?
    - How do networks of interactions actually arise?
  - CA does not consider other means of interactions
    - E.g. diffusion of properties across the environment, as in ants and field-based systems
  - CA does not consider dynamic networks
    - E.g., mobility of cells or of components across the networks, dynamic changes in the network structure